DEM 1052

Nasa CR-65346

RADIO CORPORATION OF AMERICA

MOORESTOWN, NEW JERSEY, U. S. A.

GPO PRICE $ ______

CFSTI PRICE(S) $ ______

Hard copy (HC) $5,00

Microfiche (MF) 1.25

ff 653 July 65

N66 25574

(ACCESSION NUMBER)

188

(PAGES)

CR-65346

(NASA CR OR TMX OR AD NUMBER)

(THRU)

1

(CODE)

07

(CATEGORY)

FACILITY FORM 602

2M(2-65)

March 1966

BURST TRANSMISSION
OF
PCM DATA STUDY

FINAL REPORT

CONTRACT NAS 9-4244

Prepared for:

NASA Manned Spacecraft Center
Houston, Texas

Prepared by:

Radio Corporation of America
Missile & Surface Radar Division
Defense Electronic Products
Moorestown, New Jersey

INDEX

Page

Page

FIGURES

TABLES

I. INTRODUCTION

This report discusses the study which was accomplished to determine the feasibility of applying Burst techniques to the retransmission of PCM data from multiple asynchronous telemetry links.

Burst transmission infers that data is assembled into a discrete unit and transmitted as a block. In the retransmission of telemetry data the data handling capacity of even a single input link is usually much greater than the capacity of the communication circuit over which the data is to be retransmitted, hence it is desirable to use the maximum possible capacity of the communication circuit. To achieve such high utilization when transmitting bursts it is necessary that blocks be assembled and transmitted contiguously.

After telemetry data has been retransmitted, regardless of the technique used, it is necessary that the capability exist for identifying each individual parameter. Furthermore, it will often be necessary to readout and display certain of these parameters in a periodic manner. Where PCM data is asynchronously retransmitted and then used to reconstruct the initial function, a distortion will result unless a means is provided whereby the time of occurance of each sample can be reestablished.

The burst technique studied herein are, therefore, required to perform the following functions:

- Select pre-specified parameters and assemble blocks.
- Contiguously merge these blocks into an efficient format.
- Provide for the extraction of parameters from the format (de-commutation).
- Provide for the periodic output of parameters.
- Provide a means for reestablishing the time of individual samples.

Because the distinguishing characteristic of burst transmission is that the message is composed of contiguous blocks of data, the differences which would exist between various burst systems would result from the definitions established for blocks.

A block could be defined as containing all parameters selected from a single input link. A message would then consist of the same number of blocks as there are input links and blocks would usually be rather large, containing a significant part of the communication circuits capacity.

A different definition of a block could be that it would contain parameters at a given data rate which would be transmitted so that the periodicity of the data is maintained. Blocks of this type would usually be rather small depending on the capacity of the communication circuit and the number of separate data rates from which blocks are to be assembled and many of these would be required to complete a message.

The first definition would result in the largest possible block size and the second criteria in the smallest. An investigation of burst systems based on these extremes will result in identifying the advantages and disadvantages associated with each method. A comparison of the relative merits of each will then allow a judgement to be made concerning which method would be recommended.

The two methods studied were the Periodic, small blocks of parameters at a given rate, and the Blocked, one block from each input link. Both of these methods were investigated as applied to a specific case where inputs are from Titan, Gemini and Agena link and retransmission is via a 301B(40.8KBPS) communications circuit, and for the general case where arbitrary input links and output circuits are involved.

As the result of these comparisons the Periodic Burst technique is recommended. The application of Periodic Burst to the specific case is, therefore, studied in depth and the procedure to be followed in the Periodic Burst transmission of the general case is detailed by means of logic flow diagrams and extrapolated discussions.

A demonstration of the Periodic Burst technique was conducted and is discussed.

2. PERIODIC BURST TECHNIQUE

In this section the feasibility of developing a system using Periodic Burst technique is investigated. The initial concept for a system using these techniques requires:

a) Individual blocks to be assembled from parameters selected from a single input link at a single input data rate.
b) A format to be established in which the blocks are burst transmitted at the same rate at which the data within each block occurs in order that the periodicity of the data is maintained.
c) The size of individual blocks be such that the total number of blocks required can be transmitted within the capacity of the available communication circuit.

The procedure which will be followed in this study will be to develop a Burst technique for specific system requirements and extrapolate the characteristics of that technique to a general application. The specific system requires inputs from Titan, Gemini and Agena PCM telemetry links and retransmission via the 301B communication circuit.

2.1 Basic Characteristics of the Burst Message

The Titan, Gemini and Agena telemetry formats, which were attached to the contract, have been examined and the following features of each were extracted:

Table 1

Titan		Gemini		Agena	
Bit Rate	172.8KBPS	51.2KBPS		16.384KBPS	
Sync Rate	20/Sec.	40/Sec		16/Sec	
Sample Rate	Parameters	Sample Rate	Parameters	Sample Rate	Parameters
400	20	640	6	16	112
200	20	160	3	1	144
100	37	80	9	0.2	64
40	35	40	3		
20	90	20	3		
		10	22		
		1.25	112		
		0.416	144		

A block will be assembled from parameters which occur at the same rate and during one "period" at that rate. For example, consider a block of 400SPS parameters. The data is such that all parameters to be included in a block would have occured at the rate of 400SPS and within a $10^6/400=2500$ microsecond period. It is obvious that 400 such blocks occur each second.

The total number of blocks which can be assembled from the above listed data in one second (assuming one set of blocks per input rate) is 1728.866. If these blocks are to be transmitted over a 40.8KBPS circuit each block can contain 40,800/1728.866=24.6 bits. Hence, three 8-bit words can be included in each block and 0.6 bits would be unused per block.

If it were also required that these blocks be periodically transmitted, the total slots (spaces for blocks) within the output message must be increased. Consider blocks from only the two highest rates (400SPS and 640SPS). For both of these to be periodically transmitted it is necessary that both rates be related to the number of slots as an integer. That is:

$$\frac{\text{Number of 400 slots}}{400} = \frac{\text{Number of 640 slots}}{640}, \text{ or } \frac{N_1}{400} = \frac{N_2}{640}$$

$N_2 = \frac{640N_1}{400} = \frac{8}{5}\ N_1$. The smallest number of slots which satisfy the above equation is 3200 slots per second. For transmission over the specified 40.8KBPS communications circuit, it can be seen that each such block could contain 40,800/3200 = 12.75 bits. Since all input parameters are 8 bit words, each such block can contain only one parameter (unless the word length is changed) and 4.75 bits are unused each block.

When blocks at the two different rates are merged into the format it is necessary to assure that no single slot is assigned to both rates. Hence it is necessary to determine the particular slots which will be occupied by successive blocks at each rate. The locations of these slots depend on the slot assigned for the initial block of that rate, the rate of these blocks, and the slot rate (total slots per second). Mathematically,

$$L = X + N\,(S/R), \text{ where:}$$

L = location of N^{th} block
X = location of first (inital) block
S = slot rate
R = block rate
N = successive occurance of blocks (an integer from 0 to R).

Once block rates have been specified and a tenative slot rate has been selected, it is necessary to determine if initial slot locations can be selected which will allow the transmission of blocks from both rates without interference. For the particular case being considered, a slot rate of 3200 per second has been tentatively selected to transmit blocks at 400 and 640 per second. To determine suitable initial slots it is necessary to set up the equation for both conditions and solve these equations simultaneously. Interference will occur for any slot location which will result in a solution to the simultaneous equation and, conversely, no interference will occur under the condition where a solution cannot be obtained. Determining the suitability of the 3200 slot format:

$$L_{400} = X_{400} + N_{400}\,(3200/400) = X_{400} + 8\,N_{400}$$

$$L_{640} = X_{640} + N_{640}\,(3200/640) = X_{640} + 5\,N_{640}$$

When interference occurs $L_{400} = L_{640}$, therefore:

$$X_{400} + 8\,N_{400} = X_{640} + 5\,N_{640}$$

$$X_{400} - X_{640} = 5\,N_{640} - 8\,N_{400}$$

let $X_{400} - X_{640} = \Delta_X$, then

$$\Delta X = 5\,N_{640} - 8\,N_{400}$$

Now ΔX can vary between 1 and 5. To determine under what conditions of ΔX interference will occur, each possible value can be selected and N_{400} and N_{640} will be varied to determine if the equation can be satisfied. This has been done and is tabulated below:

Table II

Suitability of Various Values of ΔX

ΔX	N_{640}	N_{400}	Interference
1	5	3	Yes
2	2	1	Yes
3	7	4	Yes
4	4	2	Yes
5(0)	8	5	Yes

It will be noted from the above table that all possible values of ΔX result in interference and therefore it is concluded that blocks at both 640 and 400 per second cannot be periodically burst transmitted in a format which contains 3200 slots per second.

The next larger number of slots which will meet the first criteria of being related to both rates as integers is 6400. It will be found that if 6400 slots per second were transmitted over a communication circuit having a capacity of 40.8KBPS, each slot could contain only 40,800/6400=6.4 bits, which is less than one word. It is therefore concluded that it is not feasible to periodically burst transmit blocks at 640 and 400 per second rates via a 40.8KBPS circuit. The original concept must, therefore, be modified.

This modified concept is - as many blocks as can be efficiently handled will be transmitted periodically and the other blocks will be non-periodically burst transmitted. It was shown above that both 640 and 400SPS blocks cannot be periodically transmitted, therefore, one of these rates will be selected as the basis for designing the format and will be periodically transmitted. The other "rate" blocks will then be non-periodically transmitted.

It can be seen that if the 400SPS rate is selected, all the other rates will be related to the slots in the format as an integer and can be periodically transmitted except two (640 and 160SPS). If the 640SPS rate were selected, three rates (400, 200 and 100SPS) could not be periodically transmitted. This seems to slightly favor the 400SPS rate as the better choice.

If the 400SPS rate were selected, each slot could contain 40,800/400=102 bits. This would allow 12 8-bit words to be transmitted and result in a potential BUE of 8x12x100/102$\gtrsim$94.2%. If, on the other hand, the 640SPS rate were selected, each slot would contain 63.8 bits, 56 bits of which could be used by seven 8-bit words, to achieve a potential BUE of 56x100/63.8$\approx$ 87.8%.

Both considerations clearly point to the 400SPS rate as the more desirable selection. Hence, the Periodic Burst system will be designed so that 400SPS data and all other rate data which are related to 400SPS as an integer, can be periodically Burst transmitted.

2.1.1 Block Rate and Maximum Length

Blocks are to be periodically transmitted at each of the above listed rates except 640 and 160SPS. The highest Periodic rate is 400 per second or one block per 2500 microseconds. Thus it is obvious that a block cannot be longer than 2500 microseconds if 400SPS data is to be transmitted. It is also obvious that if data other than 400SPS data is to be included in the Burst message, the full 2500 microseconds cannot be used by the 400SPS data. It follows then that to handle 400SPS data one block must appear each 2500 microseconds and that block must be short enough to allow other blocks to be placed between the 400SPS blocks.

2.1.2 Concept of Burst Periods and Slots

In devising a Burst retransmission format the message will be divided into periods and slots. A Burst period will be defined as the period of the highest periodic input data rate, in this case 2500 microseconds, and 400 such periods will occur each second. Each Burst period will then be divided into a number of spaces which will be called slots and which will provide a position in the message for individual blocks. One slot must be provided for each block of the message, thus the total number of slots depends on the number of blocks which are to be transmitted. Assume that, as a minimum, one set of blocks must be transmitted for each compatible rate from each input channel. Since fractions of a slot are not possible, 760 Titan blocks, 152 Gemini blocks, and 18 Agena blocks are to be included in a minimum message (640 and 160SPS rates not included) and a total of 930 slots must be provided in the format for data. In addition, at least two other slots must be provided for Sync and Timing with the result that at least 932 slots are required.

2.1.3 Discussion of Possible Period Formats

In order to provide 932 slots in a 400 period message, it is necessary that at least 132 periods be divided into three parts and the other 268 periods into two parts. All blocks from a single input channel will contain successive samples of the same parameters, hence, all blocks from a single input channel must be of the same length (minor exceptions to this will be discussed later). Considering the 400SPS input channel, it is seen that one block of this data must appear in each period. Furthermore, to maintain periodicity, that block must appear in the same relative location in successive periods in order that the separation of blocks is 2500 microseconds. Similarly, blocks from other input channels must be placed in the same relative location in their assigned periods. Two period formats would, therefore, be required. These are illustrated in Figure I.

In designing a message format it is necessary that blocks from individual input channels be placed in slots so as to maintain periodicity. When the message format is made up of different types of periods, such as those illustrated in Figure 1, it becomes necessary to simultaneously establish the location of each block and assure that the proper type period is placed in the needed location.

The second variable could be eliminated if all periods were identical. To provide identical periods and also enough slots to handle the 932 blocks, each period could be divided into three parts, thereby providing a total of 1200 slots. Each slot, in this case, would be smaller than if the period were divided in two and would obviously be capable of containing fewer parameters. However, the same number of parameters as in the two-slot case could be selected and assembled into the larger number of smaller blocks, thus allowing the same total number of data words to be transmitted.

2.1.4 Selection of the Period Format

The increased ease of programming and greater versatility which results from using identical (three-slot) periods weight heavily in its favor. Therefore, the remainder of this section will concentrate on developing a Burst system in which 400 periods of three data slots each are transmitted per second.

2.1.5 Definition of Periods, Slots and Burst Channel

In the following discussion it will be necessary to refer to individual slots, to particular periods, and to Burst channels. In order to simplify such references, each period will be identified by a number from 1 to 400, and each slot within a period by the letters A, B and C. Three Burst channels identified as A, B and C each consisting of 400 consecutive slots bearing the same letter designation exists. Individual slots can be identified by period number and Burst channel as 1A, 15B, 99C, etc.

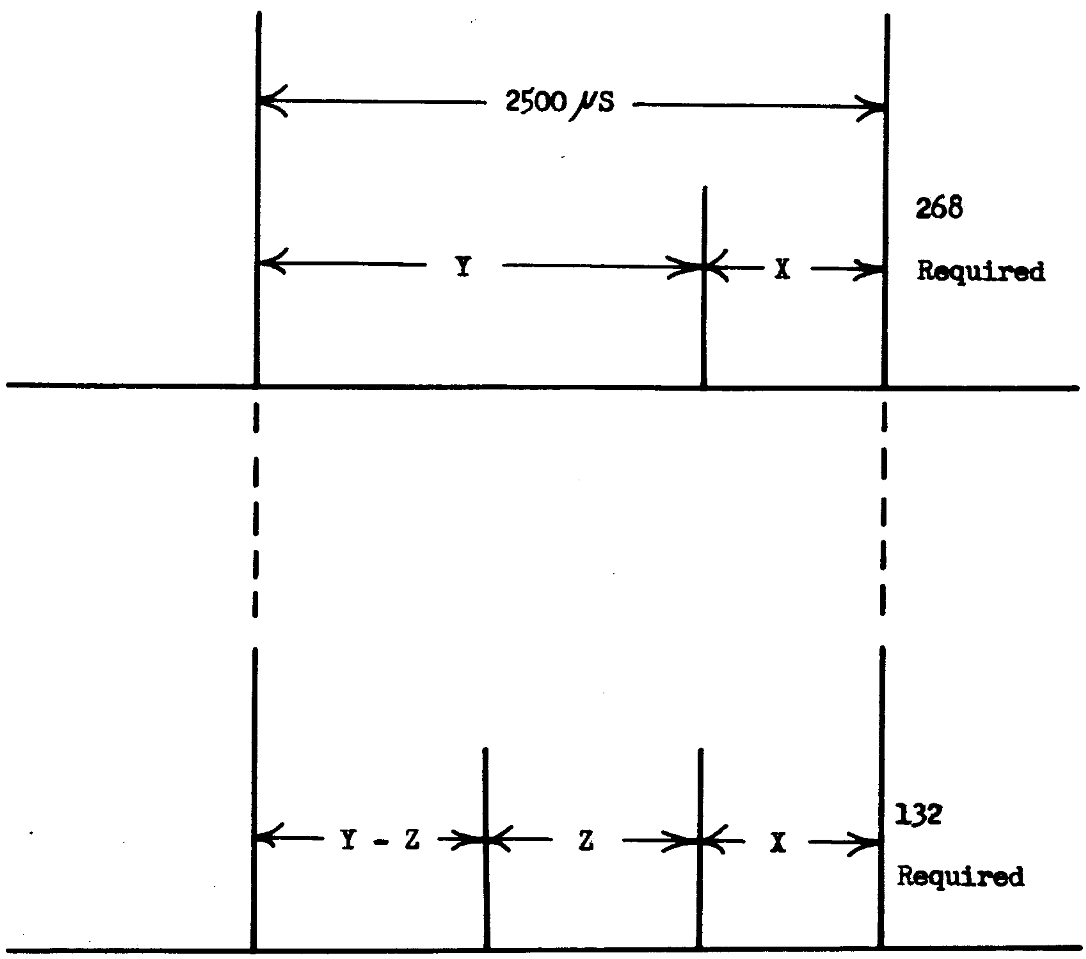


FIGURE 1. TWO-SLOT AND THREE-SLOT PERIODS

2.1.6 Relationship Between Retransmission Capability, Word Length and Block Length

The Burst system is to operate with a 301B communication circuit which is capable of handling data at the rate of 40.8KBPS(40,800 bits per second). At that rate approximately 102 bits can be transmitted within the 2500 microsecond period. All input parameters consist of 8 bit words. If these are to be retransmitted without modification (no compaction or other processing) as is assumed to be desired, each period can contain 12 words with 6 bits remaining. The total message can contain 400/96 bit periods for a total of 38,400 bits of information per message. The 301B communication circuit can handle 40.8KBPS thus the maximum bandwidth utilization efficiency (BUE) is $38.4/40.8 \times 100 \approx 94\%$.

2.1.7 Requirements for Timing and Sync

In a practical application, it will be necessary to provide Timing and Synchronization words which require slots and it may be desirable to leave a few slots vacant, hence the BUE actually achieved will be lower than the maximum calculated above.

Burst synchronization must accomplish two results. First, it must provide a means for identifying a specific part of the transmitted message so the receiving terminal can decommutate and extract identifiable data. Second, it must provide a means for correlating data within the blocks with the "time of data" from the input source.

To achieve the first purpose a synchronization word must be generated and periodically included in known slots of the Burst message. It has previously been established and reported that the optimum compromise between sync time and BUE requires a sync word of from 20 to 30 bits every 1000 to 10,000 bits of the message (1). Thus, in this message which contains 40.8 KBPS a sync word must be transmitted between 4 and 40 times per message (second), or every 10 to 100 periods.

In order that the second purpose, correlation of Burst data with link data time, can be achieved it will be necessary to establish and maintain a fixed relationship between the time of Burst sync and the time of link sync.

2.2 TIMING OF BLOCKS FROM SINGLE INPUT LINK

Assume that only one input link is present. When the master sync word is received from the input link the time of that occurence can be stored in a memory and the Burst message program can be started. At some fixed time interval thereafter a master Burst sync word will be transmitted in the first slot of the first period of the message (1A). In some subsequent slot - logically the next one - which is 1B - the stored time of link master sync will be transmitted. A clock could then cause the entire Burst program to be sequentially read-out at the rate of one period each 2500 microseconds.

If the link data rate is correct the time of occurence of each sample relative to link sync is known. Selected parameters can then be read-out of memory in blocks in which the time of all parameters are known. These blocks can be subsequently Burst transmitted in specific slots of a Burst format and the time of individual parameters can be reconstructed from the time word which appeared in slot 1B.

(1) "PCM Telemetry Synchronization" by M. Williard, 1961, Proceedings of the Nat'l Telemetering Conference.

2.2.1 Effect of Input Link Rate Errors

If, as is the usual case, the link data rate is different from nominal, the time of occurrence of all parameters will be in error from the time expected. For instance, if a particular parameter were expected 50,000 microseconds after sync and the link data rate were in error by 0.05%, the parameter would actually occur 49,975 or 50,025 microseconds after sync depending on whether the rate were high or low. A time error in the input data will carry over into a Burst message unless it is corrected and, even more critically, a time error could cause parameters in a Burst to be improperly identified.

2.2.1.1 Input Link/Burst Message Correlation at Message Rate

One possible technique for correlating a Burst message with an input link is to lock the occurrence of Burst master sync to a link master sync word. In this manner the beginning of the Burst message is accurately referenced to the input link. After that a clock in the Burst transmission terminal can be used to control the rate at which the sequencer transmits each block of data. It was previously stated that this system would read-out 400 periods per second, hence the clock would cause one period to be readout each 2500 microseconds.

As input data is received, a number of specified parameters will be selected and stored in memory. At some later time these will be read-out in precise intervals, after Burst sync, which are multiples of 2500 microseconds, i.e., 2500, 5000, 7500, etc. If the input link rate is in error by 0.05%, the delay between time of receiving a sample and the time of reading out that sample must continually change by 1.25 microseconds per period (2500 microseconds) in order to transmit that sample in the proper slot.

Figure 2 illustrates the change in delay between receiving and transmitting a parameter required by a difference between input rate and transmission rate. An input rate error of 10.0% was assumed in order that the figure could be more easily constructed and the effect would be more clearly shown. In the figure the output parameter must appear in a position which begins 2000 microseconds after the beginning of each period. Because the input parameter is occurring at a rate of 360 per second rather than 400 per second, the time between subsequent occurrences is 2750 microseconds while the occurrence of transmission slots are at a fixed period of 2500 microseconds.

Therefore, the delay in period number 1 is 1000 microseconds; in period number 2 it is 750 microseconds, etc. If the figure were extended for two more periods, it would be found that in the fifth period the input and output would occur simultaneously (zero delay) and in the sixth period read-out would occur before the data was received and, therefore, a block of "no data" would be transmitted.

The effect of a 0.05% rate error such as possible with Titan data has been magnified in the previous discussion, however a similar effect will occur and must be considered.

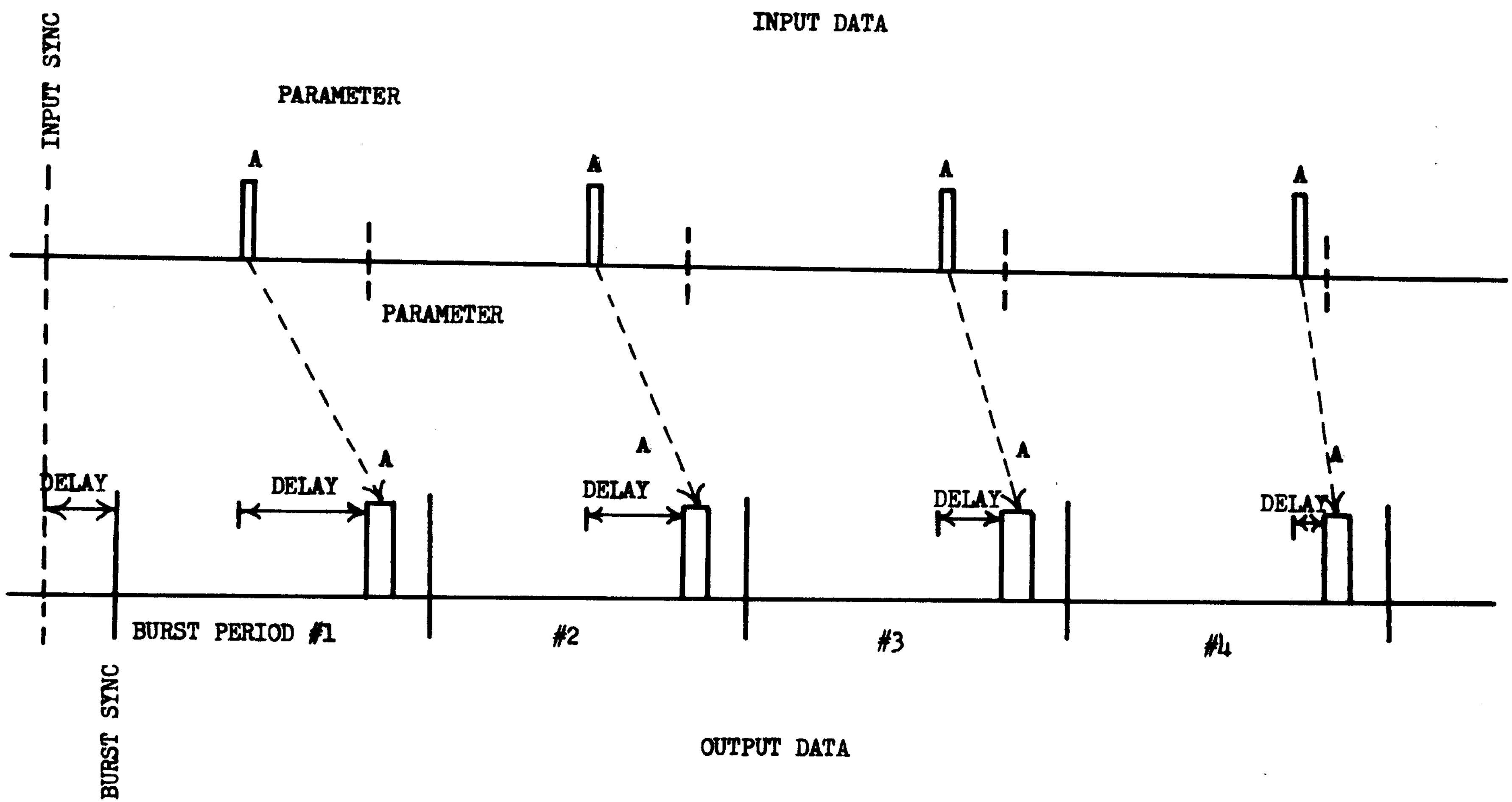


FIGURE 2. EFFECT OF DIFFERENCE BETWEEN INPUT AND OUTPUT RATES

The actual magnitude of a 0.05 % input rate error can be clearly seen by considering synchronization of consecutive Burst messages. Because the link master sync is being transmitted at the rate of 20 per second and the Burst sync at only one per second, every 20th link sync must be correlated with Burst master sync. The varying delay between subsequent samples would cause a time difference of 500 microseconds between every 20th input link sync and each subsequent Burst master sync. If this difference were not corrected the delay would continue to build up so that at the end of the next Burst message a difference of 1000 microseconds would exist, and then 1500, and so on. Eventually, the change in delay would reach 2500 microseconds and a full period would be either dropped or transmitted twice. Hence it is necessary to reestablish correlation of sync at least one each Burst message. If the time base were shifted 500 microseconds in one step to re-correlate sync, a noticeable and probably objectionable jump would appear in all visual display and decommutators may even become unsynchronized.

2.2.1.2 Burst Message/Input Link Correlation at Input Channel Sync Rate

It has already been established that a number of intermediate sync words must appear in the Burst message. Each of these words could be correlated with a link sync word. If this were done at the link sync rate of 20 per second, the maximum time base error which could accumulate would be 25 microseconds. This could be more easily compensated than a 500 microsecond error, would be less noticeable visually, and would be less likely to interfere with equipment performance.

In a system of this type, the transmission of all Burst sync words would be delayed a fixed amount from the occurrence of link sync. The occurrence of Burst sync identifies the beginning of a Burst sub-frame and initiates the first of 20 periods in that sub-frame. The subsequent 19 periods will be controlled by a precision clock and will occur in precise 2500 microsecond intervals. The 21st period (beginning of the next sub-frame) will begin when the next sync occurs. If the input rate is correct, the next sync will occur simultaneously with a 2500 microsecond interval and the Burst message will continue with no time-base error. If, however, the input rate were low, the following sync would not occur until sometime after the 20th period has been completed. This would result in a delay, during which there is no transmission, until the sync is received.

If, on the other hand, the input rate were high, the following sync would occur before the 20th period is complete. Because the occurrence of sync initiates a Burst sub-frame, the 21st period would immediately begin, thereby decreasing the length of the 20th period.

The Titan link has a specified accuracy of $\pm 0.05\%$ hence the maximum possible error in the time between successive syncs would be ± 25 microseconds. The 301B communication circuit, over which the Burst message is to be transmitted, has a rate capability of 40.8KBPS, or 1 bit per 24.5 microseconds. Thus, the shortening of the period which would result if the input rate were 0.05% high would be approximately one output bit. To assure that a high input rate would not cause a one bit loss of data, a period format could be used in which the last bit (≈ 25 microseconds) was left vacant to serve as a guard "band".

Figure 3 illustrates two periods of the Burst message, one preceding and one containing a sync word, under the three conditions of no input rate error, high input rate, and low input rate.

Figure 3-B illustrates the effect of low input rate and 3-C shows what happens to a period and the Burst message when the input rate is high.

2.2.2 Corrections for Input Rate Error

In preparing a Burst message, the location and nominal time of occurrence of individual parameters relative to link sync will be known from the basic telemetry link format. Data samples from desired parameters will be stored in memory and read-out in a block and transmitted as a Burst in a particular Burst period and slot. Thus, the identity and nominal time of occurrence of each data word within a particular block will be known. If the input rate differs from nominal, the actual time of each parameter will be in error. If some means can be found to estimate the amount of the rate error, a time correction can be computed and applied.

The Burst format which contains message sync and time of link sync in every 20th Burst period provides a method whereby the actual input link rate can be determined after the data has been Burst transmitted. If the input rate has no error the time difference between two successive "time of link sync" words will be exactly 50,000 microseconds. If, however, there is an error the time difference will be some other value, which, when compared with 50,000 microseconds, will equal the total time error that has accumulated during the previous sub-frame. A first order correction of the times of individual parameters can be made by proportionally distributing the error in accordance with the nominal time of a parameter relative to the time of the preceding link sync.

In applying this time correction in a linear manner as previously described, it has been assumed that during the 50,000 microsecond sub-frame time the rate error remained constant. Although this is a reasonable assumption when considering the data rate to be derived from a well designed and constructed oscillator operating in a controlled environment, it is not necessary to simply make this assumption. By continually comparing successive sub-frame time differences, the trend of data rate error can be determined. This will provide a means for evaluating the effectiveness of the time corrections.

2.2.2.1 Advantages

Some of the advantages of the link/rate Burst method with link/Burst sync correlation at the master link sync rate are:

1. No data sample from the controlling input link is ever "not transmitted" or is transmitted twice.
2. The maximum time error, even without correction, is limited to 25 microseconds.
3. A method of estimating the actual error and correcting the time of individual parameters is provided.

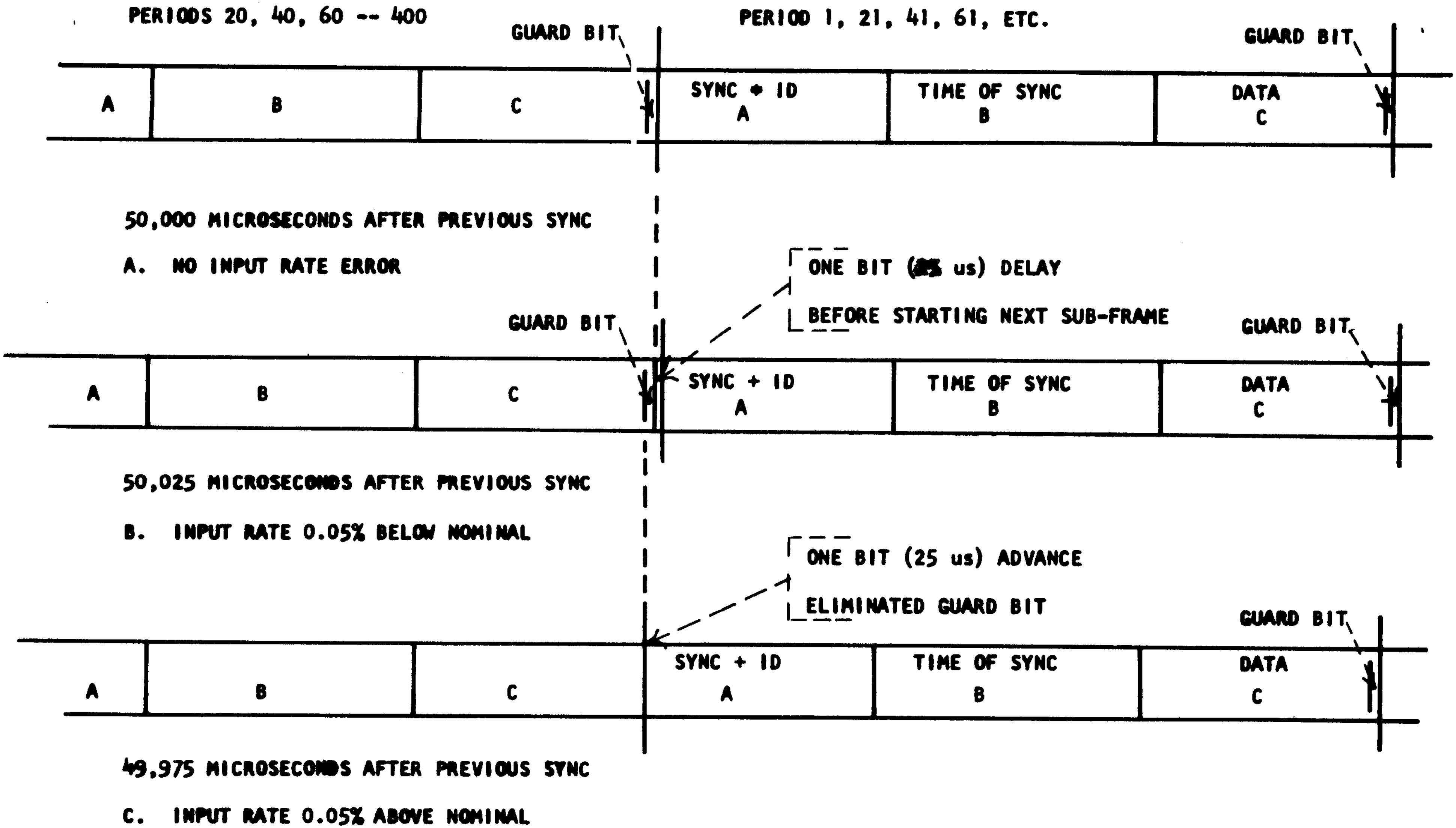


FIGURE 3 - EFFECT OF INPUT RATE ERRORS ON "SYNC PERIOD" TRANSMISSION TIME

2.2.2.2 Other Characteristics

Several other characteristics of the link/Burst sync correlation at the master link rate method (which may be disadvantages) are:

1. Absolute periodicity will not be obtained if the input data rate varies from its nominal. For example, a time base shift of 25 microseconds could occur every 50,000 microseconds. Such a shift, however, would probably be unnoticeable, or at least unobjectionable, on real-time visual displays, and because of the ability to correct time errors it could be removed in processing.

2. The 25 microsecond guard band which must be reserved because of a possible shortening of a period means that 400 bits are lost from the 40.8KB which the 301B circuit is capable of handling each second. If it were desirable to modify the input data and transmit variable length data words, this restriction would place a real limit on BUE. As presently assumed, however, this is of academic interest only, since 6 bits per period cannot be used.

2.2.2.3 Performance Uncertainty

In the periodic Burst system, each frame (interval between successive sync) would be 50,000 ±25 microseconds long. At the 40.8KBPS retransmission rate, this is equivalent to 2040±1 bits per frame.

Existing decommutators require a fixed number of bits per frame in order to perform internal checks to reliably achieve and maintain synchronization. In order to implement a periodic Burst system it is, therefore, necessary to develop a special decommutator. This special decommutator would be required to operate in a manner different than normal and would have to use techniques which have not been demonstrated. As with any development involving undemonstrated techniques, some uncertainty exists. It does appear that the required decommutator can be developed but questions concerning the performance and limitations of such a device must be established.

The reason existing decommutators require a fixed number of bits per frame, is in order that every true indication of sync recognition can be evaluated by comparison against the recognition or non-recognition of subsequent sync patterns. If a true indication is valid it is known that the next sync word must appear after a fixed number of bits have been counted. Thus, the position where subsequent sync words should be located has been established. The decommutation then "looks" for a sync pattern in that position. If such a pattern is recognized a "true" indication is given, if no recognition is made a "false" indication is presented. The decision as to the validity of sync is based on the ratio between the number of true and false indications over a selected interval.

In the periodic Burst system, however, the number of bits between successive sync patterns may vary between 2039 and 2041. Hence, a precise position where the next sync must be located is not known and the decommutator must search for sync throughout a "window" which is wide enough to include all possible locations of the sync word.

If a search throughout the "window" does not result in pattern recognition, a "false" indication will be given. Failure to obtain a true indication may be due to the initial recognition being erroneous and the sync position is not valid. On the other hand, the initial position may be valid and failure to obtain a "true" indication may be due to a signal fade or to noise. It is because of these latter possibilities that a final decision regarding sync validity requires a comparison of the number of true or false indications over some number of sync periods.

In this periodic Burst system the nominal sync periods are 50,000 microseconds long but may actually be between 49,975 and 50,025 microseconds long. If the initial sync position is valid but subsequent patterns are not recognized because of noise or fade, the window must be increased in width each subsequent nominal sync period in order that the pattern search will cover all possible positions. At the end of one period the window must cover 49,975 - 50,025 microseconds; at the end of the second period it must cover the interval of 2 (49,975) to 2 (50,025) microseconds, etc. until at the end of the n^{th} period it must cover n (49975) to n (50025).

Figure 4 is the flow chart of logic which could be applied to a Periodic Burst system decommutator, in order to achieve and maintain sync in spite of the uncertainty as to the actual sync period.

When a true indication is initially received it is assumed that the actual sync rate is nominal, and a 49,975 microsecond delay is begun. At the end of that delay a search window is opened for 50 microseconds longer than required for sync to be transmitted and an attempt is made to detect the subsequent sync pattern.

If that attempt is successful, a true indication is noted and the delay counter is recycled. If, however, sync is not detected a false indication is noted, the delay is adjusted for 49,950 microseconds, and the counter is recycled. At the end of that delay period the window is opened for 100 microseconds longer than the time needed to transmit sync, and an attempt is again made to detect sync.

If this second attempt is successful a true indication will be noted, the delay will be reset for 49,975 microseconds with a "50 microsecond" window, and the logic will be recycled. If sync is not detected, a false indication will be again noted, the delay set for 49,925 microseconds with a "150" microsecond window, and the logic will be recycled.

This process continues until either a true indication is obtained or some preselected number of consecutive false indications have occured. Each true indication is noted and the logic recycles with the delay set at 49,975 microseconds with a "50 microsecond" window.

Each false indication is noted and the previous setting of the delay is reduced by 25 microseconds while the width of the window is increased by 50 microseconds and the logic is recycled. After some preset number of consecutive false indications have occurred the equipment is judged to be "not in sync" and is switched into a search mode.

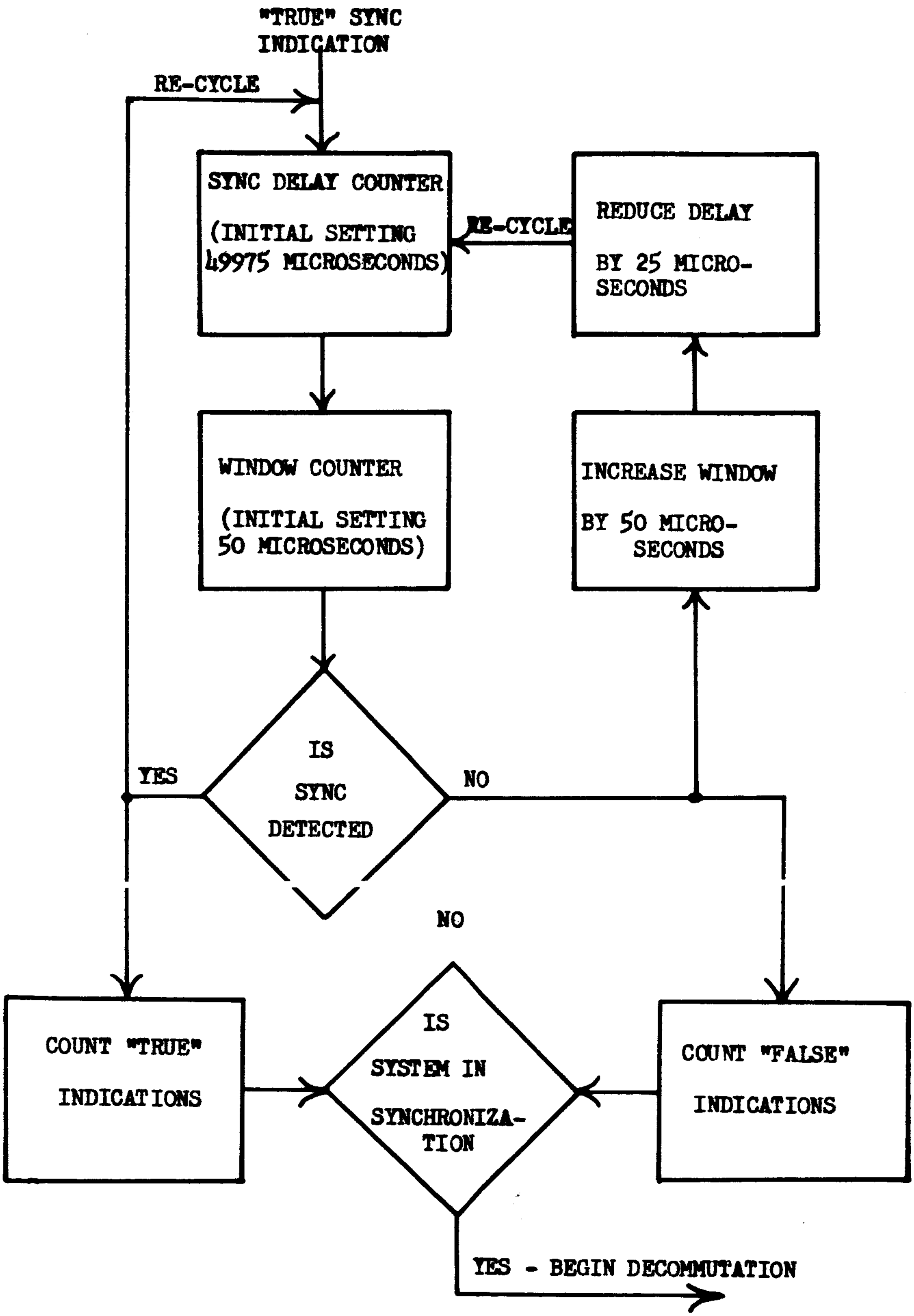


FIGURE 4. FLOW CHART OF BURST MESSAGE SYNCHRONIZER

There is little doubt that a decommutator of this type could be made to work if adequate signal is available. The uncertainty is how low can the signal to noise bebefore the decommutator ceases to work. Some of the questions to be answered are:

(a) What is the effect of continually increasing the search window?
(b) What is the relationship between the length of the sync pattern and the allowable window?
(c) How many consecutive false indications can be allowed before switching back to a search mode?
(d) How many bit errors can be allowed in a pattern before a false indication decision is made?

2.3 Timing of Blocks of Asynchronous Data

In the previous section it was established that the Burst message would be correlated with the Titan input link, thereby forming a pseudo-synchronous relationship between the Titan input and the Burst message. The effect of this pseudo-synchronous relationship on the reconstruction of Titan timing was discussed.

The Burst message is also required to contain blocks of parameters selected from Gemini and Agena links. Because the Titan, Gemini and Agena inputs are asynchronous, the pseudo-synchronous relationship between the Titan input and the Burst message will not exist for Gemini and Agena. As a result, timing of data from these links cannot be extracted from the Titan time reference and separate time references must be transmitted for each of these two links.

In this section a method for correlating the time of parameters from the asynchronous links is investigated.

2.3.1 Technique for Correlating Data from Asynchronous Links

As in the Titan link, blocks containing specific data samples will be assembled and Burst transmitted. The occurrence of each link sync forms the basis from which the identity and time of desired parameters are determined. Hence, some correlation between link sync and the Burst message must be conveyed. If the time that sync occurs on each link is stored in memory, that word can be later read-out as a Burst in some particular slot. Because the links are asynchronous, the time of occurrence of each sync word relative to each other is unknown, hence, the delay between occurrence of sync and the transmission of the time word cannot be constant as with the Titan link but must be variable so that the word can be transmitted in a particular known Burst slot.

It was previously found that the pseudo-synchronous relationship between the Titan input and the Burst message assured that no Titan block was ever "not transmitted" or was transmitted twice. Since that pseudo-synchronous relationship does not also exist for the Gemini and Agena input links, it will be possible for blocks from these links to be either "not transmitted" or transmitted twice. Although this possibility cannot be eliminated, the event can be recognized and compensating action can be taken.

If the Titan input rate is correct, the Burst message will consist of periods which continuously occur in 2500 microsecond intervals. Blocks assembled from the asynchronous Gemini and Agena links are stored in memory and read-out in pre-assigned slots. If the input rates of these asynchronous links are also correct, the delay between the time each block is assembled and the time it is read-out will remain constant. If, on the other hand, the rate of one of these links is in error, all blocks assembled from that link would have to be stored for varying amounts of time before the proper Burst slot occurs. This varying delay is cumulative, provided the rate error continues to exist, and eventually a particular block could be read-out twice before being up-dated or it could be up-dated twice before being read-out.

The Burst message will contain "time of sync" words for all links. These words will be transmitted in Burst slots which occur at the nominal link sync rate. If no rate error exists, the difference between successive "time" words will equal the period of the particular sync rate and the delay between assembling and transmitting each block has remained constant.

If, on the other hand, the difference between "time" words does not equal the nominal period of the sync rate a rate error has existed. The discrepancy between the computed sync period and the nominal sync period is equal to the change in delay which has been necessary during that time interval to assure that the blocks were transmitted in the proper slots.

From the design of the message format it will be known what nominal delay will exist between read-in and read-out of a particular block. It will also be known from the design of the system what change in delay is possible before erroneous data results. From this apriori knowledge, a block which is in error can be identified and from the history of the error between the actual and nominal "time of sync" it can be determined if the block was transmitted twice or if a block was not transmitted at all.

Once such an event has occurred, the system will again operate normally, transmitting properly identified blocks, until the change in delay again builds up to the point where an erroneous block occurs again. The time between successive occurrences of this event is equal to the nominal period of the input rate divided by the input rate error.

A similar condition would result if the Titan input rate were in error, causing the Burst periods to be transmitted at an "erroneous" rate, and the asynchronous link rates were correct. The occurrence of an erroneous block in this case would be identified from information extracted by comparing the times of successive Titan syncs.

All links could simultaneously have input rate errors and the occurrence of erroneous blocks would result from the combined effect. The identification of the event in this case requires information to be extracted from the Titan time of sync and the time of sync of each asynchronous link.

2.3.2 Limitations on Slot Location of Asynchronous Links Time Words

In selecting the slots for Gemini and Agena "time of sync" words, it is necessary to consider the maximum difference which is possible between the time of the Titan sync to which the first period of the Burst message is locked, and the time of sync of each of the other two links. The longest time difference possible would result when Titan sync occurred immediately after one of the other sync words. The maximum time difference would then be equal to the sync period at the sync rate of the particular link. The Gemini sync rate is 40 per second, hence, the greatest possible delay is 25,000 microseconds or 10 Burst periods. Agena sync rate is 16 per second resulting in a maximum time of occurrence difference of 62,500 microseconds, or 25 Burst periods. A sync word from each will always occur within those time intervals and slots must be provided, for transmission of the "time of sync" words, someplace within the calculated number of Burst periods. All possible time differences appear equally likely and, therefore, the time words could be transmitted in any convenient slots.

2.4 Design of Channel Formats

It has previously been established that the Burst message would consist of 400 periods per second each of which is divided into three slots. The message, therefore, contains three interlaced Burst channels of 400 slots per second identified as A, B and C.

The 400 periods per second were selected as equal to the highest rate data from which parameters are to be assembled into blocks and transmitted periodically. Hence, one complete Burst channel must be reserved to blocks of 400SPS parameters and channel C is so assigned. This results in two channels being available for the Burst transmission of sync, time references and other data. It seemed desirable to begin the message with a Burst sync word, therefore, channel A is assigned for this purpose and Burst sync words will appear in the first period and all subsequent ones which correspond to the Titan sync rate, in this case, periods 1, 21, 41, etc.

Because all Titan data blocks will be referenced to the time that Titan sync occurred, it is desirable to transmit the time word as soon after sync as possible. This time word will be transmitted at the same rate as Burst sync and will occupy corresponding slots in each Burst sub-frame. The Titan time word could be assigned to the previously unused channel, channel B, and placed in the same periods as the Burst sync words, or it could be assigned to the same channel as Burst sync and placed in subsequent slots. If the message were divided into only two channels, it would be necessary to use the latter type placement, however, in this case where three channels are available, Time of Titan sync is assigned to channel B in the same periods as Burst sync, i.e., 1, 21, 41, etc.

After these assignments have been made, vacant slots to which data blocks can still be assigned are available in channels A and B. There are 380 identically placed unassigned slots in each channel. Because each Burst channel can be considered as a separate identity and to do so simplifies the procedure, the formats of each channel will be independently developed.

2.4.1 Separation of Successive Blocks

For blocks of data to be periodically transmitted in a Burst channel, it is necessary that successive blocks appear in slots of the Burst channel which are separated by a fixed amount. The amount of separation required is equal to the total slots per second divided by the input data rate, or, in this case, 400 slots per second/R, samples per second. The required separations have been computed for each of the input rates from which periodic blocks are to be transmitted and are listed in Table III.

Table III

Input Rates Vs Separation of Successive Slots

Input Rate	Slot Separation
400 SPS	1
200	2
100	4
40	10
20	20
16	25
10	40
1.25	320
1.0	400
0.416	960
0.25	1600

2.4.2 Effect of Non-Compatible Slot Separation on Burst Format

One of the implications which can be drawn from Table III is that, while successive blocks from a given input rate must maintain the listed separation, it may not necessarily always appear in the same slot of successive Burst channel frames (messages). For example, the separation of successive slots of 1.25SPS input data is 320 while the Burst channel frame contains 400 slots. Hence, if one block of 1.25 SPS data appears in the 200th slot of one frame, it would appear again 320 slots later, or in the 120th slot of the subsequent frame. While this does not violate the requirement for maintaining periodicity, it does require that different slots be identified and reserved for this data in successive frames, and therefore, the problem of designing the Burst format is further complicated.

2.4.2.1 Possible Methods for Circumventing the Problem

One obvious method of circumventing this problem would be to transmit blocks, from this input rate, non-periodically, and if only one or two input channels at this rate were involved, might be a very practical solution. Another approach could be to increase the Burst frame time so that enough slots are provided to allow the same ones to be used for this data in successive frames. Referring to Table III again, it can be seen that if the number of slots in the Burst frame were increased to 2000, the same slots would be assigned in subsequent frames. The requirement for a Burst channel rate of 400 per second has been previously established, however, considering a frame (message) to be one second long was somewhat arbitrary, although practical, since all but three of the four lowest of the specified input rates fit perfectly and could be extended integer multiples of 1 second if to do so would significantly alleviate the problem of designing the Burst channel format.

2.4.2.1.1 The Synthetic Input Rate Method

Another technique appears to have a considerable amount of merit in that it would alleviate the format design problem without changing the definition of a frame and still have applications in other anticipated formatting problems area.

If five consecutive frames of 1.25SPS blocks are considered, starting in any arbitrary slot (200 will be used), it is found that successive blocks occur in successive slot locations 200, 520, 840, 1160, 1480, 1800, 2120. The identification of these slots in successive frames is 200, 120, 40, 360, 280, 200.

From the above, it will be noted that the occupied slots in successive frames are separated by 80 slots, which is the same as would occur if blocks from a 5SPS input rate were being transmitted. Hence, if the Burst format were designed to handle a 5SPS input rate, and five separate 1.25SPS input were used to provide data in the proper subsequent slots, a 5SPS input would have been synthesized and the data could be satisfactorily transmitted. The process of accomplishing this is to: first, assume a data rate which is related to the actual input rate as an integer (i.e., 5/1.25 = 4/1) and which also results in a slot separation which is related to the total number of slots as an integer (i.e., 400/80 = 5); second, select a suitable slot in the Burst channel and determine all subsequent slots required by blocks from the assumed input; third, synthesize the assumed input rate by placing a block from one input channel at the actual rate in the first slot, a block from another input channel of the actual rate in the next slot, a block from a third input channel in the next slot, etc. If enough inputs channels at the actual rate are not available to fill all slots at the synthetic rate, some of the slots assigned must be left blank.

2.4.3 Position of Initial Block

Part of the problem of integrating blocks from any input channel into a Burst channel which also contains blocks at other input rates is to determine the permissable slot locations for the first block, at each rate, in the Burst format. If it is assumed that a block from any input channel appeared in the last slot of a Burst frame, a block from that same input channel will appear some number of slots later, as listed in Table III. Hence, the initial block at each rate must be placed in the Burst format in a slot which is no further from the beginning of the Burst message than the separation of subsequent blocks within the input format as shown in Table II. In other words:

(1) A block of 400 SPS data must be placed in the first period.
(2) A block of 200 SPS data must be placed in either the first or second period.
(3) A block of 100 SPS data must be placed in one of the first four periods.
(4) Etc.

2.5 Development of Burst Channel Formats

In the format shown in Figure 4, it was shown that the first Burst period contains Burst Sync in slot A, Time of Titan Sync in Slot B, and a block of 400SPS data in Slot C. This meets the criteria previously expressed that a block of 400SPS data must appear in Burst slot number 1, but at the same time it is also shown that the whole first Burst period has been assigned. Hence, the only period still available for a block of 200SPS data, which meets the criteria, is number 2, and, therefore, a slot in period number 2 must be assigned to 200SPS data. Subsequent blocks of data from the same 200SPS input channel will appear in Burst periods 4, 6, 8, 10, 12, etc.

By nature of this format, periodicity between slots occur in any one Burst channel. Hence, if the first 200SPS block is placed in Burst channel "B", all blocks of that data will appear in Burst channel "B" and the slot assignments can be uniquely defined as 2B, 4B, 6B, 8B, 10B, etc. Likewise, if the first block were placed in Burst channel A, all subsequent blocks of that same data would appear in the same Burst channel. At this point in the development of the Burst message, channels A and B are identical (containing unassigned slots in all periods except 1, 21, 41, etc.) and obviously both channels are equally capable of handling 2000SPS blocks. Once an assignment is made for the 200SPS data, the two Burst channels are no longer equally available for the assignment of additional input channels and the selection of a particular Burst channel for data from a particular input channel may become important as will be shown later. For the present it will be assumed that Burst channel "A" contains the 200SPS blocks.

2.5.1 Selection of Input Channels for Burst Channel "A"

The development of the Burst message will continue by first attempting to completely fill Burst channel A. From Table II it can be seen that the higher the input data rate the more restrictive is the placement of the first block, i.e., at 400SPS only one period is allowed, at 200 two choices exist, etc. Therefore, the best approach in assigning slots is to consider the more difficult ones first. The next most difficult of the still unassigned rates is 100SPS and is the next one to consider.

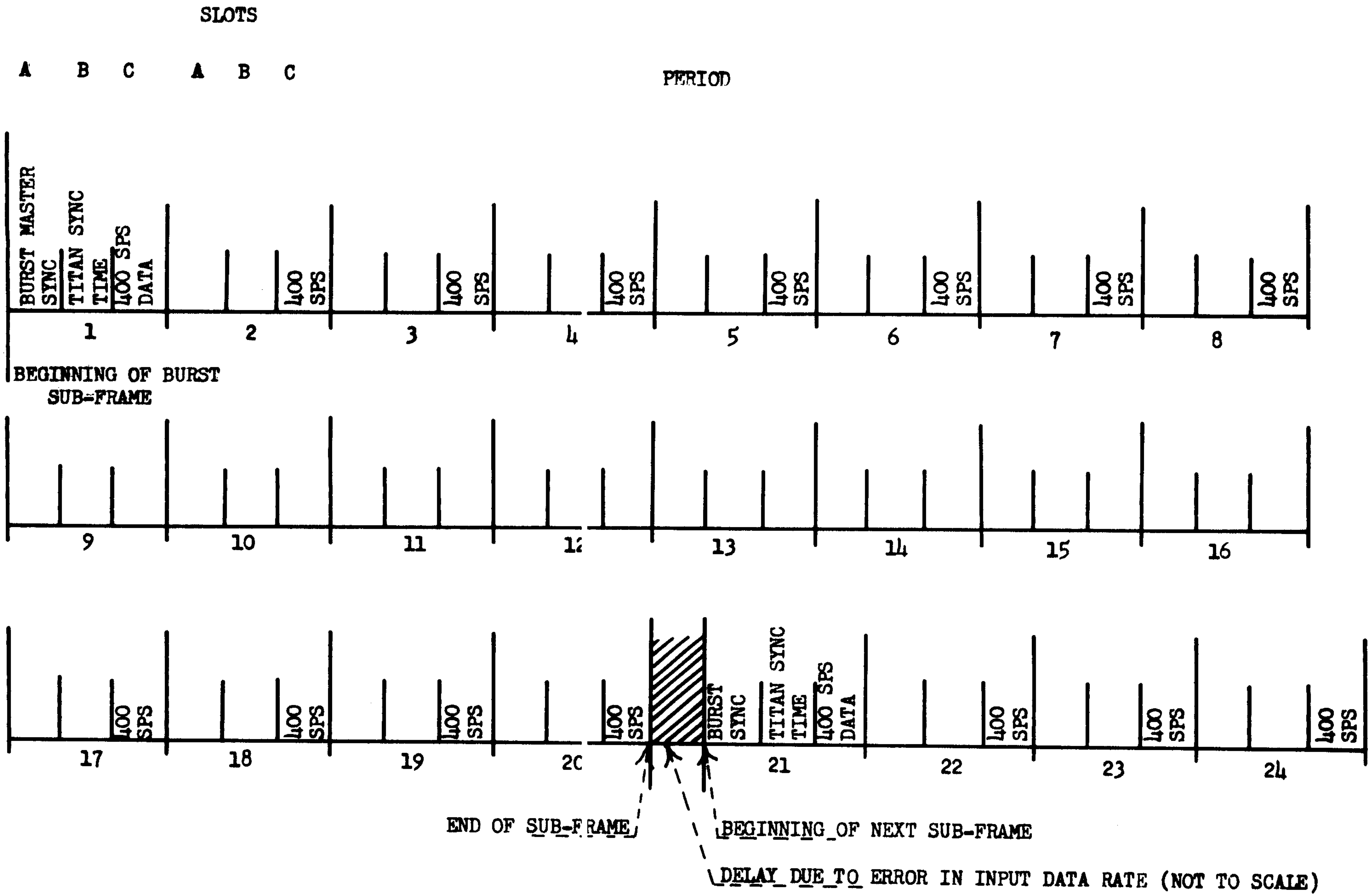


FIGURE 5. FIRST 24 PERIODS OF THE BURST MESSAGE

From the previously expressed criteria, a block of 100SPS parameters must be assigned to one of the first four periods. These blocks will be placed in Channel "A" hence the initial block of these 100SPS parameters must be assigned to slot 1A, 2A, 3A or 4A. But slot 1A has already been assigned to Burst sync and slots 2A and 4A will contain blocks of 200SPS parameters. Hence, slot 3A is the only available slot which meets the criteria and must be assigned to a block of 100SPS.parameters. Subsequent blocks of the same parameters will appear in slots 7A, 11A, 15A, etc.

Burst channel "A" now contains 200 blocks of 200SPS data, 100 blocks of 100SPS data, and 20 blocks of Burst sync. Eighty slots are still unassigned. The next highest input rate is 40SPS, two channels of which would equal the 80 slots available and, if these can be integrated into the Burst message, would result in a 100% assignment of Burst channel A.

2.5.1.1 Incompatability of 40SPS Input Channel

Placing a block of 40SPS data in the first previously unassigned slot, which is number 5, it is found that subsequent blocks fall in slots 15, 25, 35, 45, 55, etc. It is immediately apparent that since blocks of 100SPS data have also been assigned to slots 15, 35, 55, etc., the 40SPS block cannot begin in slot number 5. If the initial 40SPS block is aasigned to the next vacant slot, #9, subsequent blocks appear in 19, 29, 39, 49, etc. which again interferes with blocks of 100SPS data. The occurrence of the interference between 40SPS and 100SPS can be seen to follow a pattern such that every fifth slot of 100SPS data interferes with every second slot of 40SPS data. It will be noted that the ratio between interfering slots is 5:2 which is equivalent to the ratio of input rate, 100:40.

2.5.1.2 Integration of 20SPS Blocks

Since it has been found that the 40SPS blocks cannot be placed in the Burst channel A message, and it is desired to fully use the channel, the feasibility of integrating blocks from input channels at the next lower available input rate (20SPS) will be considered. Four such input channels would be needed to fill the 80 slots which are still unassigned. Starting with the 5th slot and then the next successive unassigned slots, it is found that the four 20 SPS input channels will result in blocks appearing in slots as shown in Table IV.

Table IV

20 SPS Input Channel	Slot Assignments
# 1	5, 25, 45, 65, 85, etc.
#2	9, 29, 49, 69, 89, etc.
#3	13, 33, 53, 73, 93, etc.
#4	17, 37, 57, 77, 97, etc.

Comparing these slots with those previously unassigned, it is found that all of these had been unassigned and, therefore, this mix is acceptable and Burst channel "A" has been 100% filled.

In reviewing the above attempts at defining a format for Burst channel A, it is noted:

(1) A successful format was developed which contains periodic blocks at input rates of 200SPS, 100SPS, and 20SPS;
(2) It was not possible to develop a format containing periodic blocks from inputs at rates of 200, 100 and 40SPS; the 100 and 40SPS blocks interfered with each other but neither of them interfered with the 200SPS blocks;
(3) It will be noted that the channels that did not interfere were related as 200:100 = 2:1, 200:20 = 10:1, 100:20 = 5:1, 200:40 = 8.1, whereas the channels which interfered were related as 100:40 = 5;2 = 2.5:1. This leads to a tentative conclusion, the input rates of data which can be Burst transmitted in periodic blocks must be related as an integer.

2.5.2 Development of Burst Channel "B" Format

The format for Burst channel "B" is yet to be defined and input channels which must still be integrated into the message occur at input rates of 40, 16, 10, 1.25, 1.0, 0.416 and 0.25 per second. Those which are likely to interfere have been determined and are listed in Table V.

Table V

Non Interfering	Interfering
40:10	40:16
40:1.25	16:10
40:1	16:1.25
40:0.416	16:0.416
40:0.25	1.25:1
16:1	1:0.416
16:0.25	0.416:0.25
10:1.25	
10:1	
10:0.416	
10:0.25	
1.25:0.416	
1.25:0.25	
1.0:0.25	

2.5.2.1 Alleviating the Non-Compatability Problems

One of the first things which can be noted from Table V is that the 16SPS input channel is the main contributer to the interference problem. If a format contained many slots and only a small percentage of them were to contain data blocks, it is possible that some arrangement of data could be found by trial and error which would allow non-compatible rates, such as 16 and 40 SPS, to both be periodically transmitted. By developing blocks from only one input channel at each of the still unused rates, a total of approximately 69 slots would be required. The 400 slots available in the Burst channel may be sufficient to allow the 69 blocks which include non-compatible rates to be periodically transmitted but only 17.8% of the Burst channel capability would be used. Such a low utilization of bandwidth would normally be completely unacceptable and, therefore, a procedure for obtaining a greater efficiency from the Burst channel is required.

An obvious way to achieve this would be to periodically transmit blocks of data from only those input channels which are compatible and to transmit blocks from those input channels which might interfere, non-periodically, in convenient slots. When this is done, a memory capable of storing all non-periodically transmitted blocks (parameters), must be provided at the receiving terminal from which parameters can be periodically readout for display by a local clock.

It is apparent that the more non-periodic blocks included in the Burst message, the larger (and more expensive) will be the required memory at the receiving terminal. Hence, it is desirable to find a means of limiting the number of non-periodic blocks within the format.

2.5.2.2 Synthesis of a Compatible Input Rate

Another approach to transmitting non-compatible data in a periodic manner and still achieve an acceptable utilization of bandwidth appears possible. Considering the 16SPS and the 40SPS input channels as the ones most likely to cause trouble, it seems that if these can be compatibly transmitted periodically, the major part of the problem will have been solved.

The two rates are related as 40:16 = 5.2. It can be recognized that if 5 of the 16SPS input channels could be combined to resemble an 80SPS channel, two compatible rates (40 and 80SPS) would result. A procedure for accomplishing this could consist of assigning slots of the Burst channel to an 80SPS input channel and then synthesizing such a channel by the proper placement of 16SPS blocks, as previously discussed in section 2.4.2.1.1.

Consider the Burst channel "B" which is being developed. This channel contains 400 slots but 20 of them(1,21,41,etc.)have already been assigned for "Time of Titan Sync". As was previously discussed, it is necessary for a block of data which is to be periodically transmitted at any specified rate, to be placed in a slot (Burst period) which is located no further from the beginning of the Burst message than the ratio of total slots to input rate, section 2.4. A block of the synthetic 80SPS data must, therefore, be placed in one of the first 5 Burst periods, i.e., (400/80 = 5). The first period has already been assigned, hence, the 80SPS channel can begin in periods 2, 3, 4 or 5. If slot 2 is selected, the 80SPS channel will have the following slot assignments in the Burst format (Burst channel"B"):

80SPS - 2,7,12,17,22,27,32,37,42,47.52,57,62,72,77,82,87,92,97,etc.

Five input channels of 16SPS data can be used to synthesize the above data, as shown in Table VI.

TABLE VI

Synthesis of 80SPS "Channel" From 16SPS Channels

Channel	Slot Assignments																				
80 SPS	2,	7,	12,	17,	22,	27,	32,	37,	42,	47,	52,	57,	62,	67,	72,	77,	82,	87,	92,	97,	102
1st 16 SPS	2					27					52					77					102
2nd 16		7					32					57					82				
3rd			12					37					62					87			
4th				17					42					67					92		
5th					22					47					72					97	

It must be recognized that the above slots are assigned to an 80SPS "channel" and even if 5 separate 16SPS inputs are not available from which data can be obtained to fill the synthetic channels, the format must be maintained, with vacant slots if necessary. For example, if only two 16SPS channels are available, the 1st and 2nd for instance, the other three slots must be provided.

This synthetic technique is similar to the one previously discussed in section 2.4.2.1.1. however, there is one major difference. The previous application of this technique was to identify slots in subsequent frames so that blocks of data at rates which are "non-compatible" with the basic Burst message can be satisfactory Burst transmissions. In this instance, however, the input rates are compatible with the Burst message and samples in consecutive frames will occupy the same slots, but the data from which the synthetic input is being developed is not compatible with some of the other input rates and the synthesis is used to obtain that compatibility.

A total of 100 slots of Burst channel "B" have now been assigned - 20 for Titan timing and 80 for data from 16SPS input channels. Therefore, 300 slots are still unassigned and can be used for blocks of data from 40, 20, 10, 1.25, 1.0, 0.25, or 0.416SPS input channels.

2.5.2.3 Inclusion of Blocks from Other Input Channels

Blocks of data from those of the remaining inputs which are compatible with the synthesized 80SPS "input" and which result in slot separations which are compatible with the Burst message (400 slots) can easily be incorporated into the message format. This can be accomplished by selecting the lowest slot still unassigned and determining the slot location of successive blocks at the rate selected. Then other input channels (of the same or different rates) are selected and the process repeated. This is the procedure which was used in developing the Burst channel "A" format when only a few high rate input channels were required to fill the Burst message. The data which will be available to complete the Burst channel "B" format, however, occurs at low rates and many separate input channels will be required. The procedure used in developing Burst channel "A", while workable, would prove awkward. Hence, the "synthetic input" method can again be used to advantage.

2.5.2.3.1 Application of Synthesis to Formatting "Compatible" Inputs

The synthetic "inputs" which can be used to aid the format design must be related to 80SPS and 20SPS as integers and must result in block separations which are related to 400 slots per second (message length) as an integer and must be less than 300, since only 300 slots remain unfilled. Rates of 80, 160, and 240 meet the first and last criteria but only 80SPS also meet the second criteria (certain lower rates also meet all criteria but since the purpose of this procedure is to decrease the number of "inputs" to be integrated into the Burst message, it will not be considered), hence, other 80SPS "inputs" must be synthesized.

In section 2.5.2.2., it was shown that all 80SPS "inputs" must begin in periods 1, 2, 3, 4 or 5. In previous parts of this development, slots in the first two periods have been assigned, hence, satisfactory starting slots are in periods 3, 4 and 5. The location of slots of 80SPS "input" data, in the Burst channel "B" format, have been determined and are listed in Table VII.

Table VII

Location of Slots from 80SPS "Input Channels

INITIAL SLOT	SUBSEQUENT SLOTS
2	7, 12, 17, 22, 27, 32, 37, 42, 47, 52, 57, 62, 67, 72, 77, 82, 87, 92, 97, 102
3	8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58, 63, 68, 73, 78, 83, 88, 93, 98, 102
4	9, 14, 19, 24, 29, 34, 39, 44, 49, 54, 59, 64, 67, 74, 78, 84, 89, 94, 99, 102
5	10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95,100, 105

Three additional 80SPS "input channels" can be synthesized from various combinations of input channels. Blocks can be included in a single synthetic 80SPS "input channel" provided they meet the criteria previously established, i.e., the rates must be related to each other and to 80 as integers and the slot separation of successive blocks must be related to 400 as an integer. Table VIIIlists combinations of input channels which are still available and which can be used to synthesize an 80SPS "input channel".

An examination of Table VIIIwill provide an indication of the combinations which can be used in synthesizing each 80SPS "input channel".

It should be noted that these synthetic 80SPS channels are used only to aid the design of the format and lose their identity once that purpose has been served.

After all four synthetic "inputs" and Time of Titan Sync have been assigned slots in the Burst channel "B", 60 unassigned slots remain. These still unassigned slots are located in Burst channel "B" of the following periods:

6, 11, 16 26, 31, 36 46, 51, 56 66, 71, 76, etc.

It will be noted that these vacant slots occur in groups of three, and the corresponding slots of subsequent groups are separated by a number of slots equivalent to that of a 20SPS input channel. These slots could, therefore, be used for the periodic transmission of blocks at compatible input rates of 20, 10, etc. There remains, however, two input channels which have not yet been used to provide blocks. These are 1.25SPS and 0.416SPS.

2.5.2.3.2 Application of Synthesis to Formatting "Non-Compatible" Inputs

It was previously discussed (section 2.4.2.1.1) how a compatible 5SPS "input" could be synthesized from four 1.25SPS input channels. If the slot separation at 5SPS is related as an integer to the separation at 20SPS, blocks from the 1.25SPS input channels can be periodically transmitted in the slots which are still unassigned. The slot separation of 20SPS blocks is 20(Table III) and the slot separation at 5SPS is 400/5=80. These two separations are related as an integer, therefore, 1.25SPS data can be periodically transmitted in the slots

Table VIII

Combinations of Input Channels to Synthesize an 80SPS Channel

Input Rates	40	20	10	1	.20	16
	2	0	0	0	0	0
	1	2	0	0	0	0
	1	1	2	0	0	0
	1	1	1	10	0	0
	1	1	1	9	4	0
	-	-	-	-	-	-
	0	4	0	0	0	0
	0	3	2	0	0	0
	0	3	1	10	0	0
	-	3	1	9	4	0
	0	0	8	0	0	0
	0	0	7	10	0	0
			7	9	4	0
	0	0	0	80	0	0
	-	-	-	79	4	0
	0	0	0	0	0	0
	0	0	0	0	0	5
	0	0	0	16	0	4
	0	0	0	16	64	3

still available. It is apparent that rather than first considering a synthetic "input" at 5SPS, an "input" at 20SPS could have been synthesized from 16 input channels of 1.25SPS data.

If a synthetic "input" of 5SPS is used, the number of 1.25 input channels which could be incorporated into the remaining 60 slots can vary in 4 input channel increments between 4 and 48, while maintaining full utilization of the bandwidth. If the 20SPS "input" were synthesized, full bandwidth utilization could be obtained in 16 input channel increments only. Since these synthesized "input channels" are artificial and used only as a convenience in designing the format, it may be desirable to synthesize both 20SPS and 5SPS channels, depending on the number of 1.25 input channels available and the number of slots which must be retained for the still unassigned 0.416SPS input channel.

Blocks from all input channels, except the 0.416SPS channel, have been incorporated into the Burst message. Blocks from that channel (0.416SPS) will now be considered.

Referring to TableIII, it is obvious that the 0.416SPS input channel does not have a channel separation which is compatible with the 400 slot message and the blocks cannot be directly included in the format. By using the synthetic "input" method it was possible to include another "non-compatible" input channel (1.25SPS), and possibly the same approach can be again used. It will be noticed that 0.416 appears to be a "rounded off" value and is actually 1/3 of 1.25. Assuming that to be the case, it follows that another 5SPS "input" can be synthesized from 0.416 input channels.

Table IX illustrates how 12 of the 0.416 input channels can be used to load a synthetic 5SPS "channel", placed in a "20SPS" group of the still unassigned slots.

Figure 6 depicts three consecutive frames of the same "20SPS" group and shows how these slots are filled by 12 input channels of 1.25SPS blocks, and 12 input channels of 0.416SPS data.

It has thus been shown that blocks from both 1.25 and 0.416SPS input channels can be simultaneously periodically transmitted in the remaining slots. Table X lists some of the combinations which are possible.

Table IX

	1st Frame					2nd Frame					3rd Frame					4th Frame					5th Frame				
5SPS FORMAT	6	86	166	246	326	6	86	166	246	326	6	86	166	246	326	6	86	166	246	326	6	86	166	246	326
0.416SPS INPUTS																									
1	6												166												326
2		86												246											
3			166												326										
4				246												6									
5					326												86								
6						6												166							
7							86												246						
8								166												326					
9									246												6				
10										326												86			
11											6												166		
12												86												246	

Table I

Input Channels	20	10	1.25	1.0	0.416
	3	0	0	0	0
	2	0	0	0	0
	2	1	8	0	0
	2	1	4	5	0
	2	1	3	5	3
	1	0	24	0	24
	-	-	-	-	-
	0	0	36	0	36
	-	-	-	-	-
	0	0	48	0	0
	0	0	0	0	144

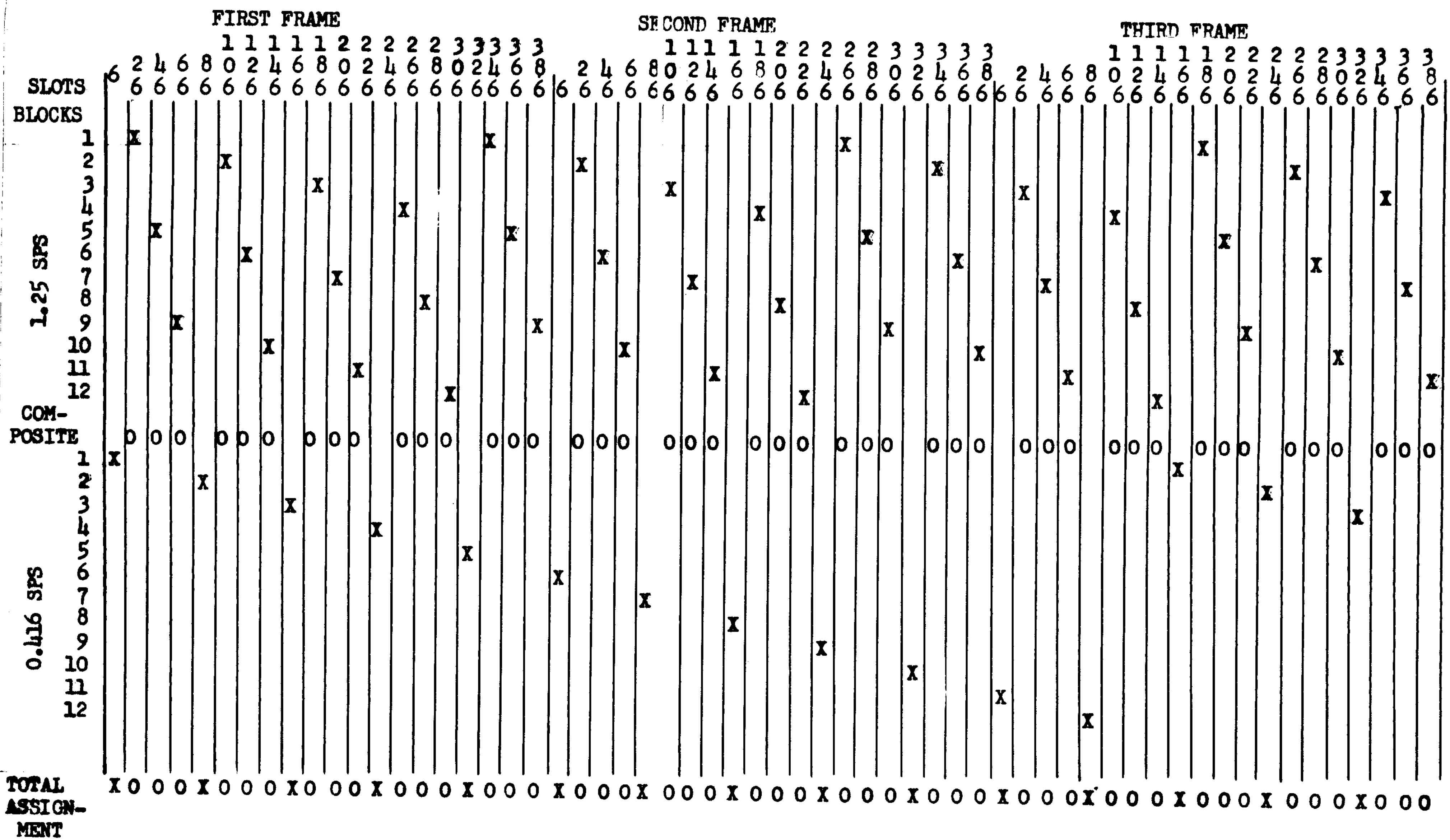


FIG. 6 SYNTHESIS OF 20 SPS RATE WITH BLOCKS OF 1.25 SPS and 0.416 SPS PARAMETERS

2.5.2.3.3 Transmission of Blocks of Non-Periodic Parameters

In section 2.1, it was decided that blocks of 640 and 160SPS parameters would be Burst transmitted without maintaining the periodicity of the parameters. The Burst channels are arranged so that all slots occur periodically and it is therefore necessary to assign suitable slots and force blocks which are non-periodically assembled to be transmitted in those slots.

A basic requirement of any retransmission system is that all data which is stored during some time interval be transmitted in an equal or lesser time interval, otherwise the amount of data in storage will continually accumulate until eventually the memory will become saturated.

The definition established for Burst periods and slots was based on the maximum transmission capacity of the communications circuit. If the periods are divided into three equal length slots - as assumed for this part of the study - the maximum amount of data which can be selected and stored between successive Bursts is limited to 32 bits - four 8 bit words. Hence, no data block can contain more than 32 bits.

A block of this "non-periodic" data could contain 1, 2, 3 or 4 words. These words could represent separate parameters or they could represent the previous 1, 2, 3 or 4 occurrences of a single parameter. A block containing separate parameters would be updated every $10^6/640 = 1560$ microseconds regardless of the number of parameters within the block. If the block contained a 1, 2, 3 or 4 occurrence history of a single parameter, it would be updated every 1560, 3120, 4680, or 6240 microseconds, respectively. As was previously pointed out, the blocks would have to be Burst transmitted within the time interval at which it is updated, to prevent saturation.

The updating intervals, computed above, correspond to rates of 640, 320, 213, and 160SPS, respectively. None of these "rates" are compatible with a 400 slot per second channel and, therefore, these blocks cannot be directly merged into the format. The problem then becomes establishing a method for assigning slots to data of these "rates".

If a rate of 640SPS is considered, it is immediately seen that while Burst slots of any channel occur only every 2500 microseconds, parameters are updated every 1560 microseconds. If blocks of 640SPS parameters are to be Burst transmitted in 400 SPS slots, it is evident that some blocks will have to contain a two occurrence history of a parameter and others could contain only a one event history of the parameter. Because of this, only two separate parameters can be handled and some blocks would contain only two words and others would contain four words. The ratio of input rate to Burst rate is 640/400 or 8/5. Hence, every five blocks must contain eight occurrences of the parameter and, therefore, three of the five blocks must contain four words and the other two blocks will contain two words each. It is obvious that in order to transmit only two 640SPS parameters, a complete Burst channel is required and the efficiency of utilization of the channel is $14/20 \times 100 = 70\%$.

If only one 640SPS parameter is to be transmitted, blocks containing three and four occurrences could be assembled. The times required to assemble these would be 4680 and 6240 microseconds, respectively, and the assemble "rates" would be approximately 213, and 160 blocks per second. If these blocks are to be transmitted at a 200SPS rate, it will be seen that some block will contain three occurrences and other blocks will contain four occurrences. Two hundred (200) blocks must then contain 640 occurrences of a parameter. The input/output ratio is 640/200 = 16/5. Therefore, 16 occurrences of a parameter must be assembled into five blocks. For this to be achieved, four blocks would have to contain three occurrences each, and the fifth block would contain four occurrences. Only about half a Burst channel would be required, and the assigned slots would have an utilization efficiency of 16/20 x 100=80%.

Blocks of 160SPS parameters could be handled in a similar fashion. At this rate, parameters repeat in 6240 microsecond intervals. Hence, a block of one occurrence of each parameter therein could be assembled each 6240 microseconds. Blocks which contain 2, 3 and 4 successive occurrences of a single parameter could be assembled each 12,480, 18,720 and 25,960 microseconds, respectively.

Because the input rate is lower than the rate at which blocks can be transmitted, it will be possible to assemble blocks of four separate parameters and Burst transmit these at a rate which is compatible with the Burst channel. A 200 block per second Burst rate could be used, however, no blocks would be available for transmission in some slots and blanks would result.

If only two 160SPS parameters were to be transmitted, these could be handled at a 100 block per second rate. Under these conditions, 160 occurrences of each parameter would have to be assembled into 100 blocks. The input/output ratio is 160/100 or 8/5, the same as previously discussed for the 640 to 400 case, and every 5 blocks would consist of 3 blocks of 4 words and 2 blocks of 2 words.

If only one 160SPS parameter is to be transmitted it would be possible to assemble 4 occurrence blocks. These would be assembled at a 40 block per second rate and they could be Burst transmitted at the same rate, thereby fully utilizing the assigned slots.

2.5.3 <u>Summary</u>

The Burst message which has been developed consists of 400 periods per second each divided into three slots, thereby forming three Burst channels identified as A, B and C.

(a) Burst channel "C" contains blocks of 400SPS parameters plus a guard bit.
(b) Burst channel "B" contains:

  (1) Time of Titan sync (20 blocks).
  (2) A synthetic 80SPS channel of:
      4 - 16SPS Agena input channels and
      1 - 16SPS "Time of Agena Sync" (80).
  (3) A synthetic 80SPS channel of:
      1 - 40SPS Time of Gemini Sync and
      1 - 40SPS Gemini Input Channels (80).
  (4) 2 Synthetic 80SPS Channels of
      40, 20, 10, 1.0 and 0.25SPS Input Channels (160).
  (5) 3 Groups of 20 SPS slots containing combinations
      of 20, 10, 1.25, 1.0 and 0.416 Input Channels (60).

(c) Burst channel "A" contains:

(1) Burst sync (20).
(2) Blocks of 200SPS "Titan" parameter (200).
(3) Blocks of 100SPS "Titan" parameter (100).
(4) A synthetic 80SPS "channel" containing 4 input channels of Titan 20 SPS parameters (80).

2.5.3.1 Burst Channel Efficiency

Burst channel efficiency is the ratio of total slots containing data to total slots available and is a measure of how well the format has been designed. There are 400 slots available in each channel but some of these are used for sync and "Timing" and are, therefore, not conveying data, hence channel efficiency is reduced.

Table XI lists the number of slots in each channel which are used for different purposes and shows the channel efficiency which results:

Table XI

Channel	Data	Burst Sync	Timing	Unused	Efficiency
A	324	20	0	0	95%
B	380	0	76	0	81%
C	400			0	100%
Total Message	1104	20	76	0	92.2%

As was pointed out in section 2.1.6, the combination of communication circuit capability (40.8KBPS) and fixed word length of 8 bits per word, results in each "block" being capable of containing 32 bits with 6 bits each period remaining. As a result of this the 1104 data block can contain 35,328 bits to provide a bandwidth utilization efficiency of 35,328/40.8 x 100 = 87.8%.

The efficiency of Burst Channel "B" can be increased by transmitting time of Gemini sync at 20 per second rather than 40 per second (transmit every other occurrence) to 85.6%. This would result in a Total Message Efficiency of 93.6% and a bandwidth utilization efficiency of 89.9%.

A further increase in bandwidth utilization efficiency would require that variable length data words be transmitted. If that were done, one bit per period would still be lost as a "guard band" and the limit on BUE would be 95.0%.

2.6 Composition of Blocks

The technique being examined herein requires that blocks of data, containing prespecified parameters which occur at known "nominal" times on a specific telemetry link, be assembled and transmitted in a "Burst". In order for this to be possible, it is obviously necessary that the parameters occur before the block is readout. Hence, there is a relationship between the time of occurrence of all parameters in a block and the time that block is transmitted.

It has already been established that the Burst message will consist of periods which occur in 2500 microsecond intervals and, for retransmission via the 40.8KBPS 301B communication circuit, each period is capable of containing 102 bits. If all data is in 8-bit words, a period can contain 12 words and 6-bits will remain. Each period must be divided into three slots which are capable of handling blocks which contain integer numbers of data words.

It has also been established that a guard slot equal to approximately 1-bit must be placed at the end of all periods which precede the Burst sync word. For convenience in preparing a message format, the "guard slot" will also be placed at the end of each period.

In section 2.1 it was shown that each Burst period could contain 96 data bits, divided among three blocks. The sizes of each block was not specified, other than to point out that all blocks of a Burst channel must be of equal length. Now, however, the size of a block will enter into the study and must be defined. The 96 data bits within a period can be divided into equal length blocks of 32 bits each (4-8 bit words); or in other ratios, to the point where one block contains 10 words (80 bits) while the other two blocks contain 1-8 bit word each. The time required to readout a block on the 301B communication circuit is 196 microseconds per word. Time to readout blocks of various sizes from 1 to 10 words long is given in Table XII.

Table XII

Word/Blocks	Readout Time (microseconds)
1	196
2	392
3	588
4	784
5	980
6	1176
7	1372
8	1568
9	1764
10	1960

For this study it will be assumed that all data blocks will contain 4 words, therefore all data slots must be 784 microseconds long.

It has previously been established that slot "C" must always contain a block of 400SPS parameters plus a guard bit, hence slot "C" must be 809 microseconds long and must begin 1691 microseconds after the beginning of each period.

Slot "A" contains either data blocks or a block of burst sync plus sub-frame identification. This slot begins at the start of each period and is 784 microseconds (32 bits long). Five of these bits must be used to identify one of 20 sub-frames, hence, the burst sync pattern is limited to 27 bits.

Slot "B" will then be 2500-(784+809) = 907 microseconds (37 bits) long. This slot starts 784 microseconds after the beginning of each period.

In the Burst format previously developed, the Burst sync slots occur some fixed delay (time) after occurrence of Titan master sync. The time at which all Burst slots (beginning of block readout) occur are fixed to that event. It is, therefore, possible to establish a chart showing the times after beginning of Burst sync when each data slot begins. Once this has been done, a basis for identifying the specific Telemetered parameters which could be included in a particular slot will exist. Table XIII lists the time of the first 20 slots and identifies the contents of each (the first two slots contain Burst sync and a "Time of Titan Link Sync" words which are developed in the equipment and are not carried over from the telemetry channel).

Because Burst sync has a fixed relationship to time of Titan Sync, the time of slot occurrence limits the times at which parameters to be included in each slot must occur, otherwise the parameters may have occurred both before and after the sync word.

Table XIII

Times and Contents of Slots of the Burst Format

Slot No.	Contents	Time After Burst Sync Beginning
1 A	Burst Sync	0
2 B	Time of Titan Sync	784
2 C	T-400	1691
2 A	T-200	2500
2 B	Time of Agena Sync	3284
2 C	T-400	4191
3 A	T-100	5000
3 B	Time of Gemini Sync	5784
3 C	T-400	6691
4 A	T-200	7500
4 B	T-40	8284
4 C	T-400	9191
5 A	T-20	10000
5 B	T-20	10784
5 C	T-400	11691
6 A	T-200	12500
6 B	G-0.416	13284
6 C	T-400	14191
7 A	T-100	15000
7 B	A-16	15784

2.6.1 Titan Blocks

2.6.1.1 400 SPS Blocks

Slot 1C of the Burst message is the first slot which contains a data block. The parameters in that block are from the Titan 400 SPS input channel. Table XIV identifies all 400 SPS parameters in the Titan message and tabulates the time after Titan sync at which each occurs.

Table XIV

Time of First Occurance of Each 400 SPS Parameter

Parameter Number	Location Word & Syllable	Time After Sync (Microsec.)
177	1-1	44.9
178	1-2	89.8
179	1-3	134.7
180	2-1	196.4
181	2-2	241.3
182	5-1	650.9
183	5-2	695.8
184	5-3	740.7
185	6-1	800.4
186	6-2	845.3
187	9-1	1148.3
188	9-2	1193.2
189	9-3	1238.1
190	10-1	1283.0
191	10-2	1344.7
192	13-1	1754.3
193	13-2	1799.2
194	13-3	1844.1
195	14-1	1905.8
196	14-2	1950.7
---	---	---
177	17-1	2544.9

If a block of 400 SPS Titan contains two or more parameters, these could be numbers 177 and 196 and 1905.8 microsecond of elapsed time could be required to load the memory. From Table XIV it can also be seen that only 594.2 microseconds are available between the occurrence of parameter number 196 and the subsequent reoccurrence of parameter number 177. The maximum size block which can be readout within that time interval can contain only three words (Table XII) and even this is possible only by carefully selecting readout time relative to Titan sync. Such a limitation would be unacceptably restrictive on the system and cannot be tolerated.

To overcome this limitation, a pair of identically organized memories called Continuous Selection memories, can be used to select the 400 SPS parameters. One of these would be connected to the input link in order that parameters can be selected and stored, while simultaneously the other memory is connected to the Burst sequencer which can cause the previously stored parameters to be readout. In this manner the memories are automatically available for loading, during a 2500 microsecond period, and then for readout during the subsequent 2500 microsecond period.

Figure 7 illustrates the manner in which continuous selection memories are used to assemble and readout blocks of 400 SPS data while maintaining identity and time correlation of all parameters.

Several characteristics of this Burst technique which effect figure 7 are:

(1) The occurrence of Titan master sync forms the basis for developing and controlling the Burst message and for controlling certain internal functions needed to assemble the data.

(2) It can be seen that when the Titan sync occurs, Memory #1 is connected to the input and #2 is connected to the output, if this condition does not already exist and 2250 microseconds later these memories are interchanged. During the 19 subsequent intervals the memories will be automatically connected to the input or the output each 2500 microseconds. The reason that the memories are interchanged 2250 microseconds after sync is to assure that the switch-over takes place after parameters #177 and #192 have been loaded into one memory during the same period and before parameter #177 re-occurs. The time selected is arbitrary but is more than adequate to allow for input rate errors and switching time. One of the two memories is designated Number One, and will be connected to the input each time the Titan Sync is received. Switch-over may actually occur when Titan sync is initially received, however, at all subsequent occurrences of sync memory #1 will already be properly connected and the only effect of sync on these memories is to establish switch-over time. If the input rate is nominal (no error) the intervals between switch-overs will be constant at 2500 microseconds, however, if there is an input rate error the time of switch-over at the end of the interval during which sync occurs will be advanced or extend in proportion to the rate error.

(3) The parameters in the block which is transmitted in slot "C" of Burst period #1 occurred within the 2500 microsecond interval after link sync. Those in slot "C" of period #2 occurred between 2500 and 5000 microseconds after Titan sync, etc.

2.6.1.2 Blocks of 200 SPS Parameters

From Table XII it can be seen that blocks of Titan 200SPS parameters are transmitted in slots 2A, 4A, 6A, etc. Slot 2A occurs in the Burst message 2500 microseconds after sync and slot 4A occurs 7500 microseconds after Burst sync.

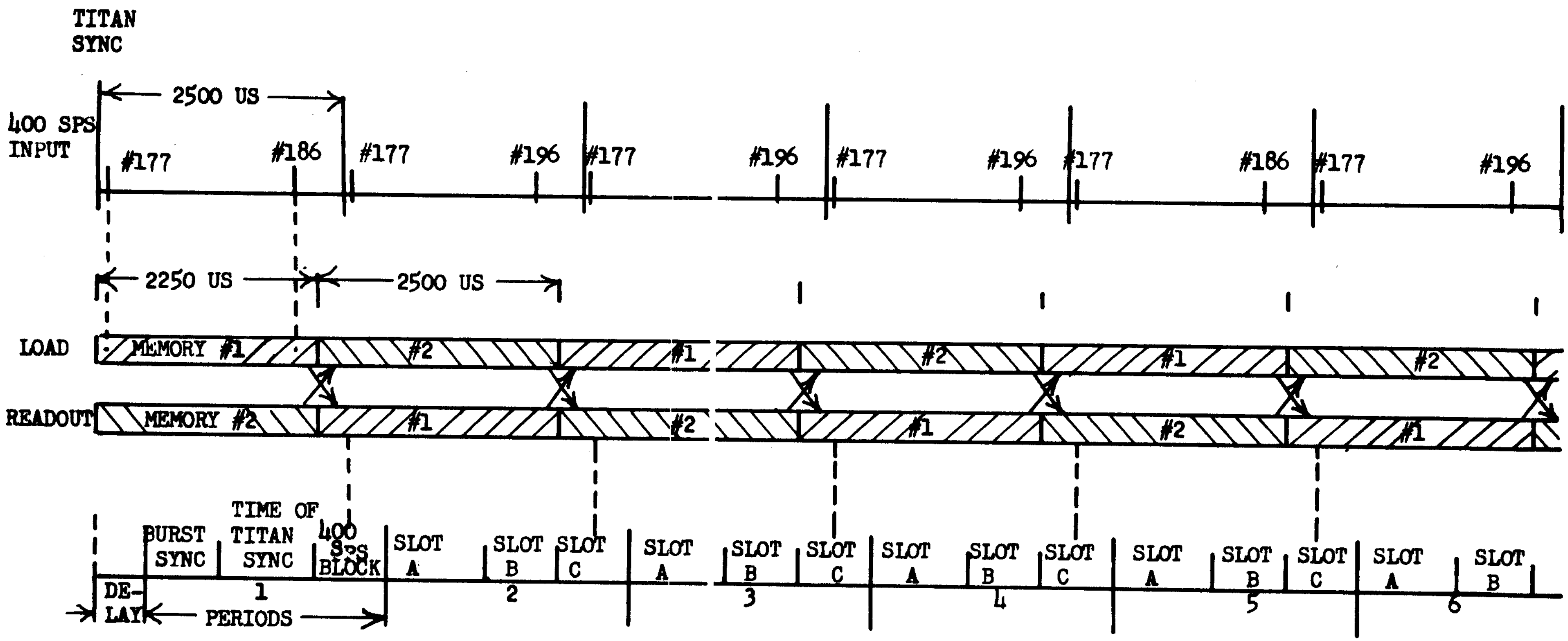


FIG. 7 - APPLICATION OF CONTINUOUS SELECTION M MORY FOR BURST TRANSMISSION OF 400 SPS PARAMETERS

From the basic Titan format, it is found that there are twenty 200SPS parameters, identified as numbers 157 thru 176. Parameter 157 occurs first 347.9 and again 5196 microseconds after Titan sync. The first occurrence of parameter 176 is 4180.3 microseconds after Titan sync. The elapsed time required to receive and store a block of 200SPS data containing both of these parameters is 3832.4 microseconds, regardless of how many other 200SPS parameters are included and if dual alternating memories are not used, this block must be readout before parameter 157 occurs again. Hence, readout must occur within the 1015.7 microsecond interval between the first occurrence of parameter 176 and the second occurrence of parameter 157. In order for slot 2A to occur at the correct time, it would be necessary to properly delay the start of the Burst message from the arrival of Titan sync.

Since slot 2A must occur at least 4180.3 microseconds after Titan sync, and 3284 microseconds after the Burst message begins (Burst sync) a delay of at least 1679.7 microseconds would be required.

The maximum size of the block which could be transmitted in slot 2A is 1005.7 microseconds - the time interval between occurrence of parameter 176 and the subsequent parameter 157. TableXII shows that a block of 5 words can be readout within that interval. This will probably be adequate, however, a larger size block could be transmitted and the time of transmission could be freed from a direct dependence upon the delay between link sync and the start of the Burst message, if dual identically organized memories were used as with the 400SPS data.

If "Continuous Selection" memories were used, one of these would be connected to the input at the time Titan sync is received, and interchanged each 5000 microseconds thereafter.

2.6.1.3 Blocks of 100SPS Parameters

In the format, the first block of 100SPS data is transmitted in slot 3A, 5000 microseconds after Burst sync, and the next subsequent block of the parameters occur in slot 7A which is transmitted 15,000 microseconds after Burst sync.

The Titan format contains 36 parameters, numbers 121 to 156, at this 100SPS rate. The first occurrence of parameter #121 is 241.3 microseconds after Titan sync. The other parameters of this group occur in sequence until the first occurrence of parameter #156 takes place 9482.4 microseconds after Titan sync. Hence, the elapsed time for consecutive parameters to be received is 9241.1 microseconds, which is the time interval required to assemble a block containing these parameters. Obviously, if the 121st and the 156th parameterwere not both included in a single block, the elapsed time would be decreased, however, the capability of including these two parameters must be provided in the event they are needed.

In comparing the times of transmission of various slots, the time of arrival of various parameters which may be placed in these slots, two features are evident:

> Slot 3A must contain a block of 100SPS data (sec. 2.5.1), yet this slot occurs before about half of all 100SPS can be updated in the time since Titan sync. A block of 100SPS data transmitted at this time could contain some parameters which actually occurred since Titan sync and other parameters which occurred before Titan sync. It would be possible to determine which ones occurred before and which ones after sync, however, to accomplish this would require that some of the parameters be referenced to the time transmitted in slot 1B and others to be referred to the time transmitted in slot 381B. Furthermore, it would necessitate carefully selecting readout time to assure no parameter is being updated while it is being readout.
>
> It is, therefore, concluded that this input channel can best be handled by identically programmed dual alternating memories, in which the time of all parameters are referenced to the preceding time of Titan sync in slot 381B.

2.6.1.4 Blocks of 20SPS Parameters

Five 20SPS input channels, one of which is "time of Titan sync", are used to fill the Burst channel "B" (sec. 2.5.1.2). The four other 20SPS inputs all contain telemetered parameters and are assigned initial slots of 5B, 9B, 13B and 17B. Table XV lists each of these slots, the time after Titan sync that each is transmitted, and the identity of all 20SPS parameters which have been updated since the last Titan sync and which could be assembled in a block to be transmitted in each slot.

Table XV

Initial Slot	Time of Slot Since Burst Sync	20SPS Parameters Which Have Been Updated since Titan Sync
5B	10,784 microseconds	#1-#13, #61,66,71.76,D1
9B	20,784 "	#26, 62, 67, 72, 77, D2
13B	30,784 "	#38, 63, 68, 73, 78, D3
17B	40,784 "	#50, 64, 69, 74, 79, D4
Not Updated		#51, 60, 65, 70, 75, 80, D5

From the above table it can be seen that blocks of data can be assembled and transmitted in the assigned slots from parameters which have occurred since the last Titan sync unless one of the last 15 parameters is included. In that case, either the time of that one parameter must be reconstructed separately from the others, or identically organized dual alternating memories must be used, and all data in a block formed in this manner referenced to the previous sync time.

2.6.1.5 Summary

The time interval which may elapse before all parameters to be assembled in a block are received, could be as long, after Titan sync, as a period at the input data rate. In designing the Burst message format, it is necessary to efficiently use the available bandwidth, and it is desirable to eliminate as many variables as possible in order to simplify the process of designing the format. Hence, it is desirable to be able to design the format without considering the actual time of occurrence of individual parameters.

To provide the desired freedom in designing a Burst format and still assure the "time integrity" of all parameters within a block, a continuous selection (alternating identical organized memory s ections) technique can be used. When using such a technique a particular memory section would be connected to the input upon receiving a link sync word. A fixed-time interval thereafter, which is a function of the input data rate, the memory sections are interchanged. This results in all parameters in a block which is being read-out, having occurred during the previous input data cycle (see Figure 7).

2.6.2 Blocks of Asynchronous Data

Blocks of parameters from the asynchronous Gemini and Agena links also must be assembled and transmitted in the Burst message. Unlike the conditions which exist with the Titan link, there is no fixed relationship between the occurrence of Gemini and/or Agena syncs and the time that blocks of data from these links are transmitted. Hence, a variable delay between occurrence and transmission of parameters (blocks) must be provided, and an uncertainty as to the actual time of occurrence of individual parameters could arise unless care is taken to assure that such confusion does not exist. In Figure 8 four 16SPS Agena parameters are shown as an example to illustrate how this confusion could occur.

Four 16SPS parameters, identified as 1, 2, 3 and 4, and occurring 1100, 3900, 11,400 and 18,900 microseconds, respectively, after Agena sync, are to be transmitted as a block in Burst slot 7B.

Figure 8A illustrates the condition when Agena sync nearly coincides with the Burst sync. The previous Agena sync (#1) which occurred 62,500 microseconds earlier, is also shown. The times of arrivals of the four parameters relative to each occurrence of Agena sync are identified as is the position of two Burst slots, numbers 382B and 7B.

After Agena sync number 1 occurs, parameters 1, 2 and 3 occur and are loaded into memory. Burst slot 382B follows and the memory are unloaded as a block. Subsequently, parameter #4 occurs and is stored in memory. Later, after Agena Sync #2, parameters #1, #2 and #3 appear again and are stored in memory. Burst slot 7B follows and the memories are read-out a s a block containing parameters 1, 2, 3 and 4. It will be noted that parameter #4 occurred prior to Agena Sync #2 while the other three parameters occurred afterwards. The result is that the times of individual parameters must be reconstructed from two different references (or if from the same reference, addition is required for some parameters and subtraction from others.)

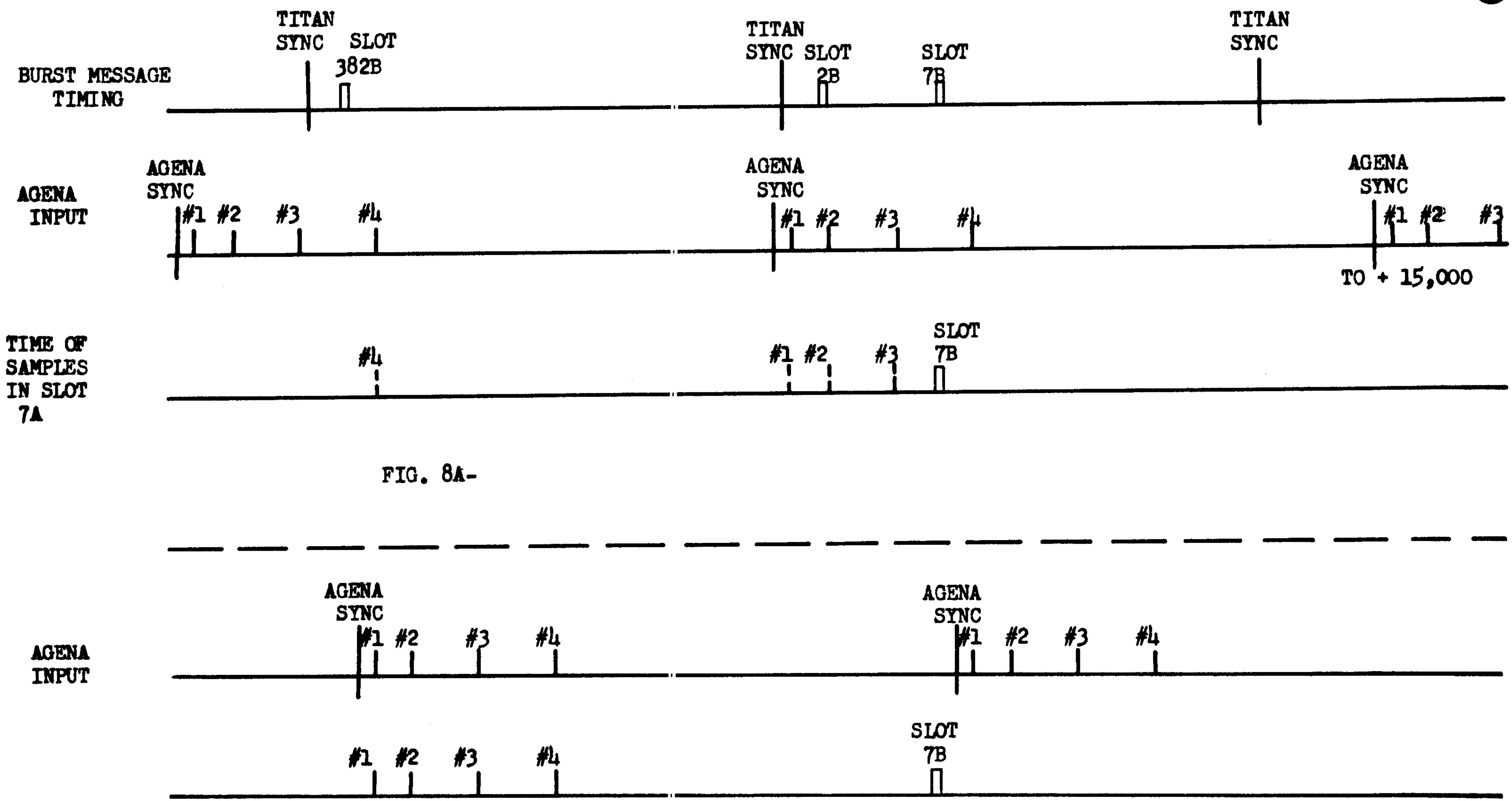


FIG. 8A-

FIG. 8B

FIG. 8 EFFECT OF TITAN AND AGENA ASYNCHRONOUS RELATIONSHIP ON TIME OF SAMPLES IN A BLOCK WITH SINGLE CYCLE MEMORY

Figure 8B shows what happens when Agena Sync and the Burst slot occurs almost simultaneously. In this case, all parameters read-out in a block will have occurred prior to that Agena sync and after the previous Agena sync.

A general statement can be made: If the Burst slot occurs after all parameters of a block have been received and before any of them can be updated (re-occur), all parameters in the block will have occurred during the same period (of the input data rate).

Inasmuch as no control can be exercised over the time of occurrence of Agena sync relative to Titan sync, and therefore the Burst message, the system must expect conditions where the time of slot occurrence does not meet the criteria stated above. Hence, a technique which can assure the time relationship of all parameters within a block, must be provided. Such a technique is the "continuous selection" method.

2.6.2.1 Continuous Selection Memory

Assume that the Agena parameters shown in Figure 8 are to be assembled via continuous selection memories and Burst transmitted in the slots shown.

When Agena sync #2 occurs, the connections to the memories are interchanged. The memory section which had been connected to the input and in which the parameters #1, #2, #3 and #4 had been stored in the interval between Agena sync #1 and #2 is now connected to the output. When slot 2B occurs, the time of Agena sync #2 is Burst transmitted. Later, when slot 7B occurs, the parameters stored in memory are read-out as a block. These parameters occurred after Agena sync #1 and before #2, hence the time of each can be re-established for the Time of Agena sync #1, transmitted in Burst slot #377B.

A condition which exists with both Gemini and Agena which requires special system features to assure the proper identification and timing of individual parameters. This condition results from lower rate data originating on sub-commutated channels and gives rise to a requirement for transmitting a means of identifying sub-frame sync.

The Agena format contains three sub-frames, A, B and C. Sub-frames A and C re-cycle together once each second (each 16 mainframes) and sub-frame B re-cycles once each 5 seconds (every 80 mainframes). Every 80th mainframe all three sub-frames begin a cycle. Hence, if a number from 0 to 80 were transmitted, it would identify successive sequences of the B sub-frame and every 16th number (0, 16, 32, etc.) would identify the beginning of sub-frames A and C.

Figure 9 illustrates the formation of Burst periods which contain "time of sync" words from three asynchronous input links.

Figure 9A illustrates the periods which contain Burst sync and "Time of Titan Sync" words. The "Time of Titan Sync" is transmitted in hours, minutes, seconds and microseconds.

Burst periods which contain "Time of Agena or Gemini Sync" and identification of the mainframe sequence are shown in Figure 9B. The period contains 102 binary bits, one of which is required as a guard band and 64 must be used for blocks of selected parameters in Burst channels "A" and "C". Of the 37 bits remaining, 7 must be used to identify between 1 and 80 sequences of the Agena mainframe, thereby leaving 30 bits available for timing. Timing can, therefore, be transmitted in seconds and microseconds. Because of the relationship between time of Titan sync and Time of Agena Sync (see Section 2.3.2), the hour and minute part of the time word appears as part of the "Time of Titan Sync" word. Also, the "second" (time) part of the Agena Time would be the same as the "second" (time) portion of the Titan Time Word which contains the proper hour and minute words.

Figure 9C illustrates all other periods.

The Gemini format contains a prime subframe operating 1/4 as fast as the mainframe and sub-commutated at 8/1 and 24/1 ratio. Ninety-six (96) mainframes must be sequenced in order to sample all data inputs. It is, therefore, necessary that the master frame sequence identification number be capable of identifying between 1 and 96.

2.6.2.2 Sub-Commutated Data

In section 2.5.2.3.2, blocks of sub-commutated parameters at 1.25 and 0.416SPS from the Gemini link are assigned slots in the format. Figure 6 illustrates this format.

One input channel of 1.25SPS parameters are assigned slots 26 and 346 in the first Burst frame, slot 266 in the second Burst frame, slot 186 in the third Burst frame, etc. Slots assigned to blocks from this channel in one frame are assigned to blocks from another input channel in subsequent frames, i.e., slots 26 and 346 in the second Burst frame are used by blocks assembled from different 1.25SPS input parameters. Hence, it is necessary to establish which of 24 sub-frame sequences and which of four master frame cycles the parameters in a particular block were selected. Figure 10 illustrates the occurrences and assembly of blocks of sub-commutated data.

Block 1 is assembled of parameters which occurred during the 3rd master frame cycle of the 1st sub-frame cycle; Block #2, of parameters which occurred during the 1st master frame cycle of the 3rd sub-frame cycle; Block #3, of parameters received during the 3rd master frame cycle of the 3rd subframe cycle.

For clarity and convenience of discussion, it is shown that all parameters within a block occurred during a single cycle of the master frame. In actual practice, however, the parameters within a block could be selected over many cycles of the sub-frame. The limitation is that the parameters assembled into a block must be consecutive. As a practical consideration, it is desirable that the elapsed time between the first and last parameter should be kept as short as possible in order that the time of each parameter can be more easily and accurately reconstructed.

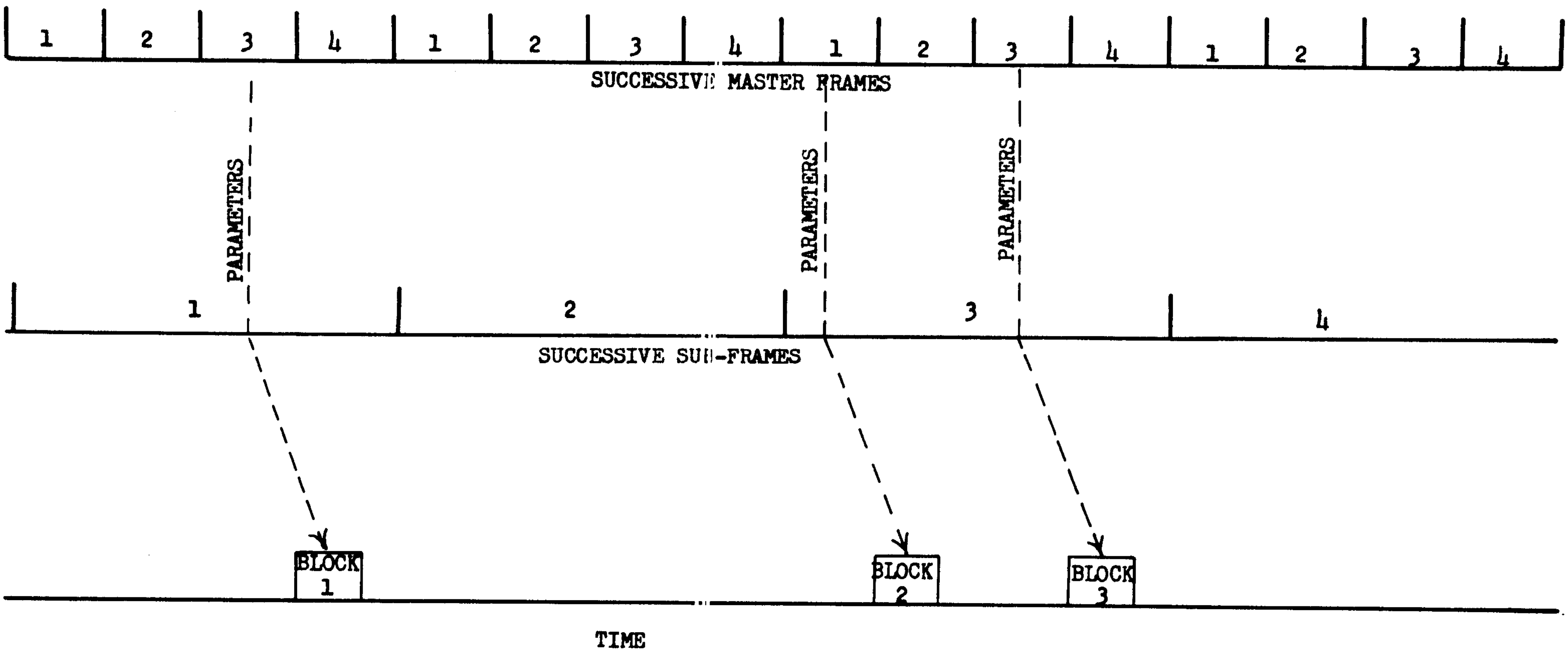


FIG. 9 - ASSEMBLY OF BLOCKS OF GEMINI SUB-COMMUTATED PARAMETERS

In referring back to the format, Figure 6, it can be seen that all three of these blocks will be transmitted in the same slots but during different frames of the Burst message. Logic can be designed which can select the optimum slots for each block and read these blocks out into the Burst message. In addition, however, it is necessary that information concerning the identity of the block which is transmitted in a particular slot and the basics for reconstructing timing, also be conveyed in the message.

One way to convey identification is to include an address word in each block. This address word could simply be a number from 1 to 12 (4 bits) which identifies which block memory is being transmitted. Individual parameters could then be identified from the apriori knowledge of which parameters had been programmed into that memory.

The time of occurrence (nominal) of each parameters will be known from the Gemini format relative to the time of the 1st of 24 sub-frame syncs. But 96 Gemini master time of sync words are transmitted in the Burst message before all sub-frames are cycled. The 96 time of sync words, in addition to the method used to readout blocks in particular Burst slots, causes an uncertainty as to which time word should be used as a reference in re-establishing time. Thus, there is a possibility of 96 ambiguities which must be resolved.

These ambiguities can be eliminated if a means of identifying a particular master frame "time of sync" word in relation to a prime sub-frame cycle, is provided. Every fourth mainframe sync is followed by a prime sub-frame sync word which includes a 5-bit address which identifies which of 24 sub-frames is being cycled. (From the input link sub-frame cycle identification an unambiguous identification can be established.) The number (1-96) can then be included as part of the Gemini "Time of Sync" word (see Figure 11).

To re-establish the time of individual parameters, the process is:

(1) Identify a particular block of sub-commutated parameters from the 4-bit address word which is included in the Burst. The cycle of the master and sub-frame in which the parameter occurred is known from the data selection program.

(2) Find the "Time of Master Sync" word which contains the address word which identifies the cycle of the master frame (1-96) which corresponds to the parameters in the block.

(3) Read the "Time of Sync" word and add the nominal time of occurrence of each selected parameter.

(4) Compare the interval between successive "Times of Sync" with 25,000 microsecond (the nominal sync rate) and correct the time associated with each parameter (see Section 2.2.2).

2.7 Implementation Requirements

Figure 10 is a simplified block diagram of the "Periodic Burst" transmission sub-system. This figure illustrates the various data and other signal sources used to select parameters, assemble blocks, and generate the Burst Transmission format.

The Sequencer contains a programmable memory, a Timing Sub-Unit, and Buffers necessary to input and output signals and data.

2.7.1 Periodic Burst Transmission Sub-System

2.7.1.1 Message Program

Prior to an operation a program to control the readout of the various blocks must be developed and stored in the Sequencer memory. This program will consist of the addresses of the memories which are to be readout, arranged in the sequences that readout is desired. When the Burst message begins, signals (pulses) will be applied to the memory at the times each subsequent block is to be read, and the memory will advance, one slot for each pulse, in the same manner as a "step switch". In this manner the order of the blocks are controlled by the program but the time of occurrence of each is controlled by the timing sub-unit.

2.7.1.2 Message Slot Timing

It has been previously established, that the Burst message will contain 20 sub-frames, at the Titan sync rate, each of which is divided into 20 periods. The first 19 of these periods are precisely 2500 microseconds long, controlled by the clock reference, and the 20th is 2500 $\pm$ 25 microseconds long depending on the actual Titan sync rate.

The first slot of each sub-frame contains a Burst Sync Block which is made up of two parts. One of these is the burst sync pattern and the other is the sub-frame identification number. When the Sequencer addresses the readout of a Burst Sync Block, the sync pattern is extracted from the sync generator and attached to the sub-frame identification extracted from the Sequencer. The block is then transmitted.

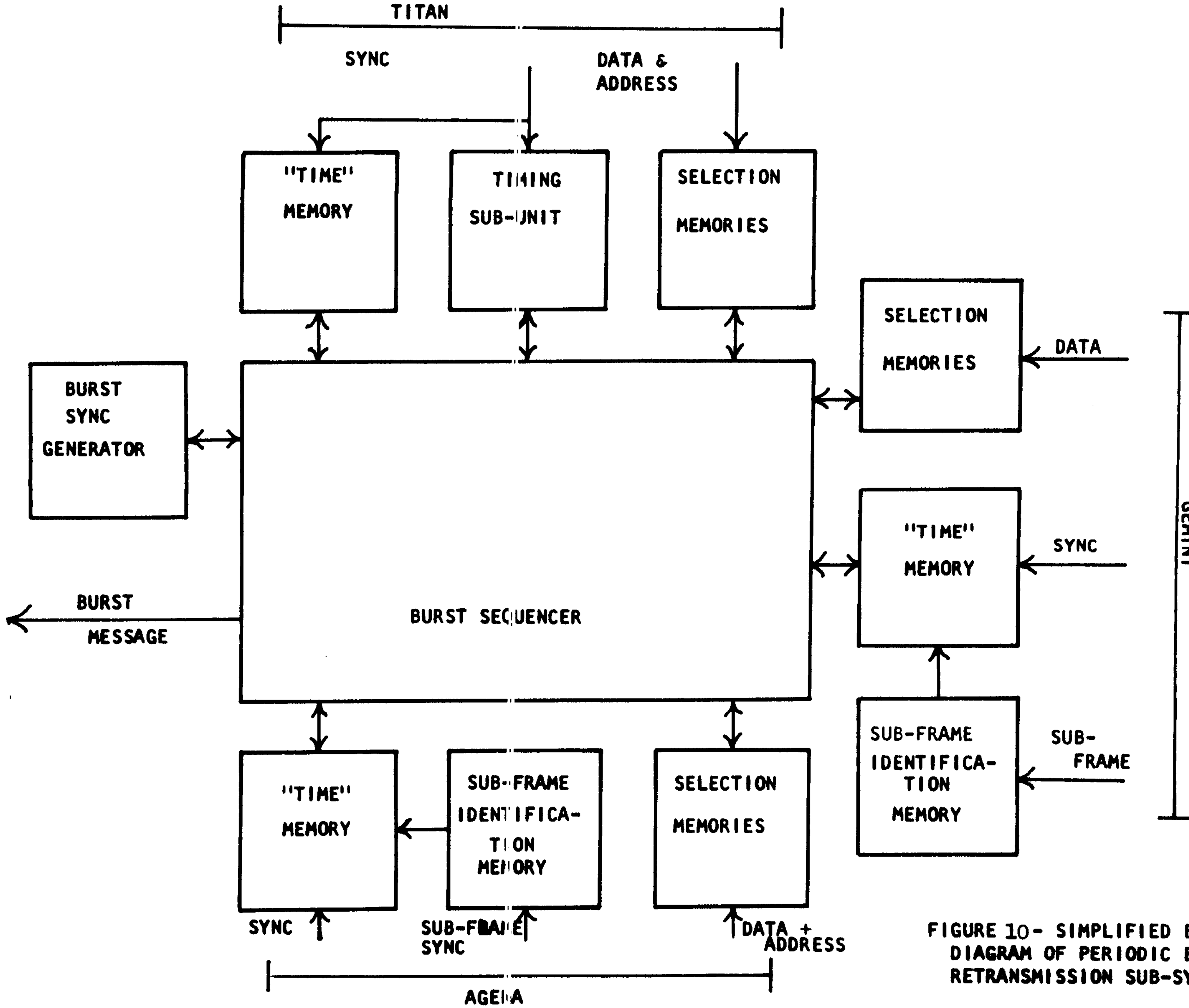


FIGURE 10- SIMPLIFIED BLOCK DIAGRAM OF PERIODIC BURST RETRANSMISSION SUB-SYSTEM

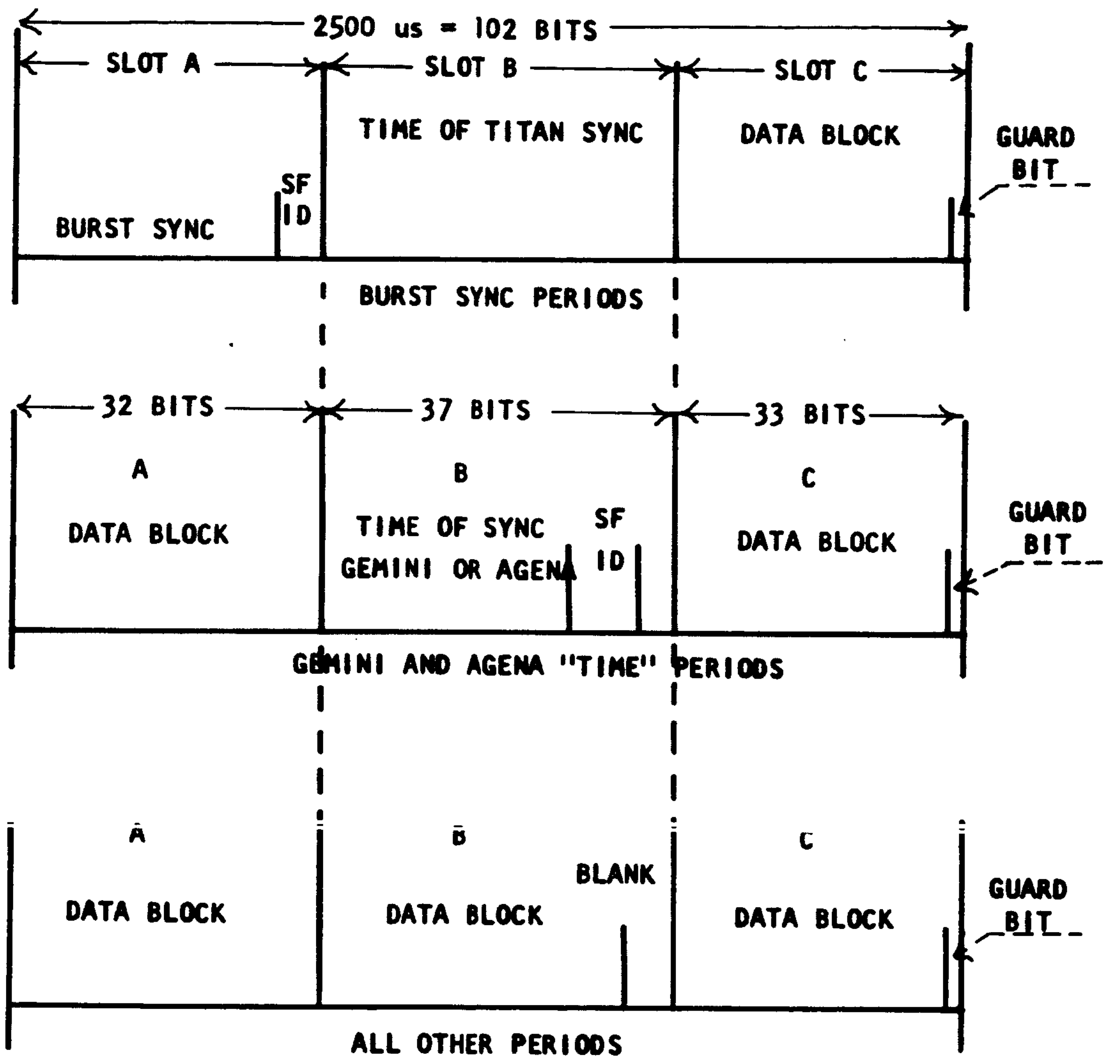


FIGURE 11- BURST PERIOD FORMAT AND UTILIZATION

2.7.1.3 Timing Sub-Unit

The Timing Sub-Unit generates the signals which cause the Sequencer to cycle through its program as described in Section 2.7.1.1. The signals are derived from a reference frequency which is accurate to at least 1 in 10^8, and therefore precision control of the format is achieved.

In previous sections, burst transmission of blocks assembled from three simultaneous input links was discussed. One of these links is selected to control the burst message. The occurrence of sync from that link enables the burst message which is then pseudo-synchronous with the control link. Under some conditions, it may be necessary to transmit the burst message when the control link input is not present. Features which provide this capability are included.

The order in which specific blocks are to be transmitted is stored in a program in the sequencer and defines the burst message format. A signal applied to the Sequencer causes the format to be advanced on step and to read-out the next block of the program. The timing sub-unit generates the signals which establish the times that subsequent blocks are addressed by the program.

The format is divided into sub-frames which contain twenty burst periods. Each period is divided into three unequal length slots - A, B and C. The times that these slots occur are controlled by the Timing Sub-Unit, hence this unit must generate signals at unequal intervals corresponding to the lengths of the various slots. Furthermore correlation must be established and maintained between the individual blocks of the format and the corresponding slots established by the Timing Sub-Unit. To achieve the pseudo-synchronous relationship between the control link sync and the burst message, it is also necessary that the transmission of the first block in a sub-frame be correlated with the occurrence of sync on the control link.

Figure 12 is a Simplified Block Diagram of the Timing Sub-Unit. The lengths of the three slots are established by delay counters, identified as A, B and C, corresponding to the blocks which will be transmitted while these counters are operating. These counters are preset so that the intervals between signals are equal to the time required to transmit the corresponding block and, in this particular case, are 784, 907 and 809 microseconds, respectively.

These counters operate in sequence. A signal applied to any one of them simultaneously starts that counter and causes the Sequencer to transmit a programmed block. When the present count is reached, a signal to control the subsequent event is generated and used to control the next counter and block read-out.

To assure that correlation exists between the program and the Timing Sub-Unit, a flag is extracted from the sequencer whenever the next block to be read will be in channel A. This flag is used to gate the "A" counter and inhibit all others.

After a full burst period (slots A, B and C, in sequence) has been generated one of two possible events occurs depending on the operating mode which has been selected. The alternatives are discussed as applied to the individual modes.

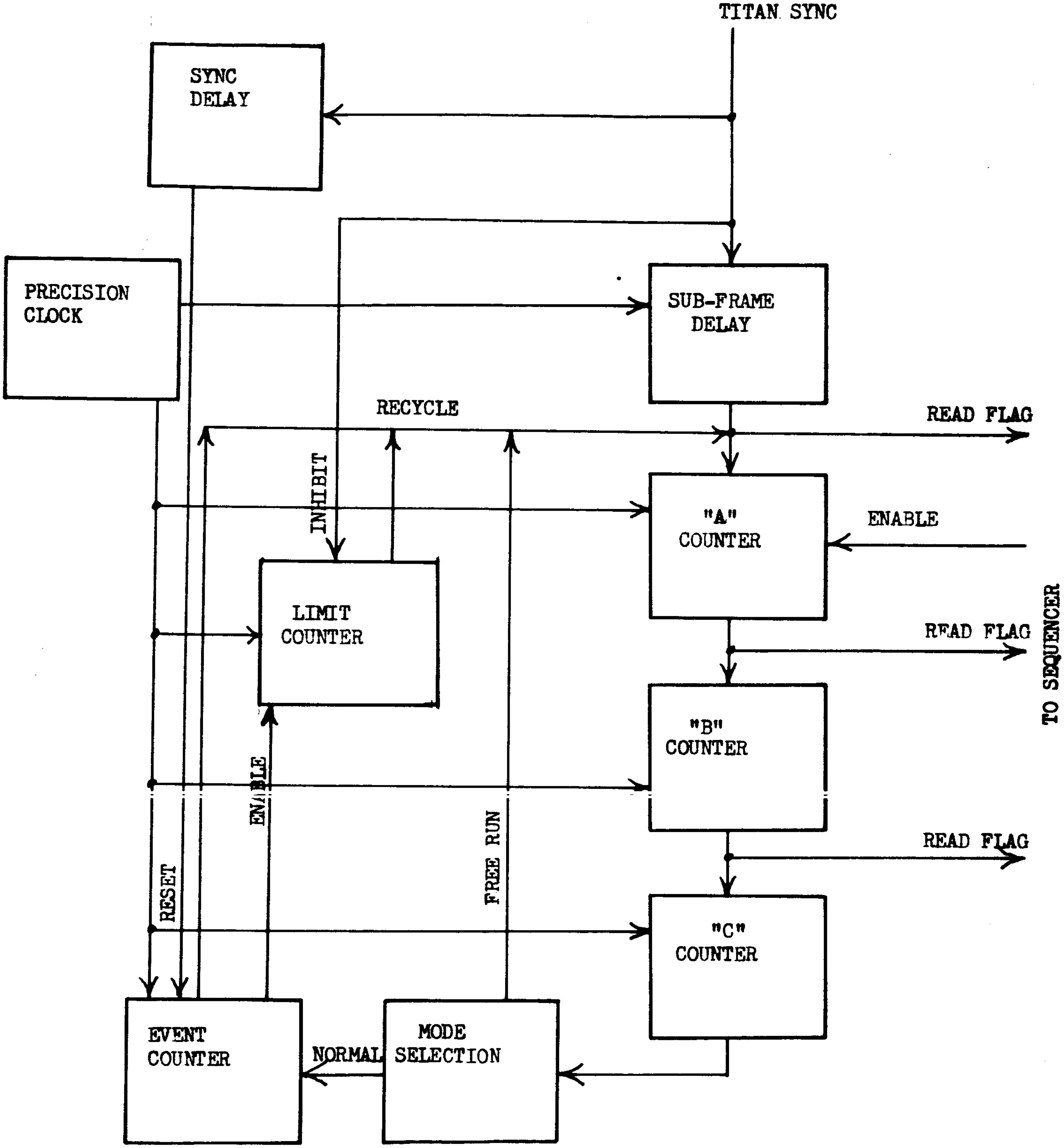


FIG. 12 - SIMPLIFIED BLOCK DIAGRAM OF TIMING SUB UNIT

2.7.1.3.1 Normal Mode

The Normal Mode is the one which has been developed throughout this study and is characterized by the pseudo-synchronous relationship between the control link (Titan) and the burst message.

To obtain this relationship requires that a burst sub-frame be started when, and only when, Titan sync occurs. This is accomplished during pre-transmission preparations by adjusting the program to the beginning of a sub-frame in order that the occurrence of a Titan sync flag will cause the first block of a sub-frame to be addressed at the same time that the "A" counter is started. After count "A" is completed counters "B" and "C" sequentially operate with the output from each starting the next and causing the program to advance and transmit the next block.

A burst period is complete when counters A, B and C have completed their preset counts. Because only the first period of each sub-frame is started by Titan sync and 20 periods must be generated during one sub-frame, a means for generating the intervening 19 periods is required. This is achieved by an event counter which allows 19 consecutive occurrences of the "C" counter output to be applied to the "A" counter thereby initiating another burst period, and which prevents the 20th occurrence from recycling. In this way the start of each sub-frame is locked to the occurrence of Titan sync but intervening periods are precisely controlled by the Timing Sub-Unit.

The operation of the counters assure that the event counter is reset to one (1) exactly 50,000 microseconds after a Titan sync started a sub-frame. The tolerance of the Titan input is given as $\pm$ 0.05%, hence a subsequent sync flag can occur anywhere between $\pm$ 25 microseconds of the reset time.

If the Titan sync occurs early, the "A" counter is started and the next block is addressed by the program before the "C" count is complete. This could reduce the length of the C block by as much as 25 microseconds (1 bit) but would not cause a loss of data since that bit is provided as a guard "band" to allow this condition to be tolerated. The completion of the "C" count would then simply reset the event counter to one (1) and thereby prepare it for operation at the end of the next period. If the next Titan sync is late, the event counter would have already been reset and the message would have been interrupted awaiting the Titan sync.

If the next Titan sync is not recognized for any reason the burst message would not restart and the interruption would continue until a Titan sync does occur. During such an interruption no message would be transmitted and all real time data would be lost. To prevent such a condition when the Titan sync is temporarily not recognized, an Automatic coast feature is provided.

Automatic Coast is a form of logic which limits the interruption between sub-frames to 25 microseconds even if the subsequent Titan sync does not occur. This feature is accomplished by a 25 microsecond counter which is started at the instant the event counter is reset, unless this counter had been inhibited by the occurrence of Titan sync during the previous 25 microseconds. The occurrence of Titan sync after the limit counter is started will terminate the count and reset the unit. If the termination does not occur a signal will be generated at the end of 25 microseconds which will cause another sub-frame to be transmitted.

Once a burst message has begun normally, the Auto Coast will maintain transmission through any failure of Titan sync. Operation in Auto Coast can be continued indefinitely. Although the resulting burst message resembles a normal message, it differs in several important characteristics:

(1) The "Time of Sync Blocks" will contain "no data" and therefore Titan sampling times must be reconstructed from the last Titan "time" block before the Auto Coast becomes operative. Time uncertainties therefore increase with the number of sub-frames generated by Auto Coast.

(2) Sub-frames occur at 50,025 microsecond intervals. This is the interval which would result if the sub-frames were generated by sync flags when the Titan sync rate was 19.99 per second; ie. 20-0.05%. Since it is probable that the actual Titan sync rate would be different than this value, an accumulating error such as discussed in Section 2.1.1 would occur. This accumulating error would cause the pseudo-synchronous relationship to cease to exist.

If after some number of sub-frames have been generated by the Auto Coast feature the Titan sync is again recognized, the message can either be continued in Auto Coast or it can be returned to normal operation.

If operation is continued in Auto Coast, several characteristics will result:

(1) The error will continue to accummulate and eventually a sample will not be transmitted.

(2) The "time of sync" blocks will contain time words. These can be used as references for re-establishing sampling times. However, only relative time could be established for supercommutated parameters, although absolute time can be derived for samples at prime or subcommutated rates.

If Titan sync re-occurs before the accummulated error exceeds the time of the last slot in the sub-frame (slot 20C, 40C, 60C, etc.), the system can be automatically returned to normal operation. The block being transmitted in this slot would be stopped at the instant sync occurs and at least part of the samples would be unusable. However the subsequent parts of the message would be correct and no further loss of data would occur.

If however, the accummulated error exceeds the time of the "0" slot, the correlation between the block being addressed by the program in the Sequencer and the slot being generated by Timing Sub-Unit would be lost. This would result in the subsequent message being scrambled and would prevent decommutation and extraction of useful data. To overcome this it is necessary to either continue in Auto Coast, or to re-synchronize the Sequencer with the timing sub-unit. To re-synchronize would require that the program be stopped at the beginning of the next sub-frame and await the occurrence of the next Titan sync before continuing. This would result in an interruption of the message, and a loss of data, of up to one sub-frame.

2.7.1.3.2 Free Run Mode

The Free Run Mode is provided so that the burst message can be transmitted when the control link (Titan) is not present. This mode is similar to Auto Coast except: (1) Free Run is manually selected, (2) previous transmission in the Normal Mode is not necessary, and (3) the sub-frame interval is 50,000 microseconds.

The sub-frame rate in the Free Run Mode is probably more nearly the actual rate than that used in Auto Coast, hence the "error" will accummulate more slowly. Even though the actual rate could be higher or lower than the nominal and a block could either be transmitted twice or not transmitted at all, such an occurrence would happen less often than in Auto Coast. This is an advantage which may make it desirable to switch to Free Run whenever Titan sync is missing for more than a few sub-frames.

2.7.2 Periodic Burst Decommutation and Parameter Extraction Sub-System

After the Burst message is received, it is necessary that blocks be identified and separated, and individual parameters must be extracted and time tags attached. The process is essentially a multi-stage decommutation.

2.7.2.1 First Stage Decommutation

After synchronization has been achieved in a manner similar to that described in section 2.2.2.3, this decommutation stage performs the function of extracting and identifying individual blocks. The output of this first-stage decommutator consists of separate blocks from which individual parameters must still be extracted.

Figure 13 is a flow chart illustrating the internal functions of the first-stage decommutator. It can be seen that the decommutation logic is similar to the logic involved in assembling the message (see section 2.7.1.3.) Because there are only three different types of blocks to be identified and these occur in a fixed known order, the first stage decommutation process is straightforward and uncomplicated, and the equipment logic is simple in concept and could be simply implemented.

The operation of this decommutation stage is as follows:

(1) The Burst message consists of 400 periods each divided into three slots. The configuration of the different periods are shown in figure 11. The first period of each Burst sub-frame contains Burst sync and decommutation must begin with one of these periods.

The "A" slot of each period contains 32 bits. In Burst sync periods the first 27 form a sync pattern and the remaining 5 bits identifies which of the 20 Burst sub-frames is being received. Immediately after the sync pattern is recognized, the next 5 bits containing a number between 0 and 19 are extracted and stored in memory.

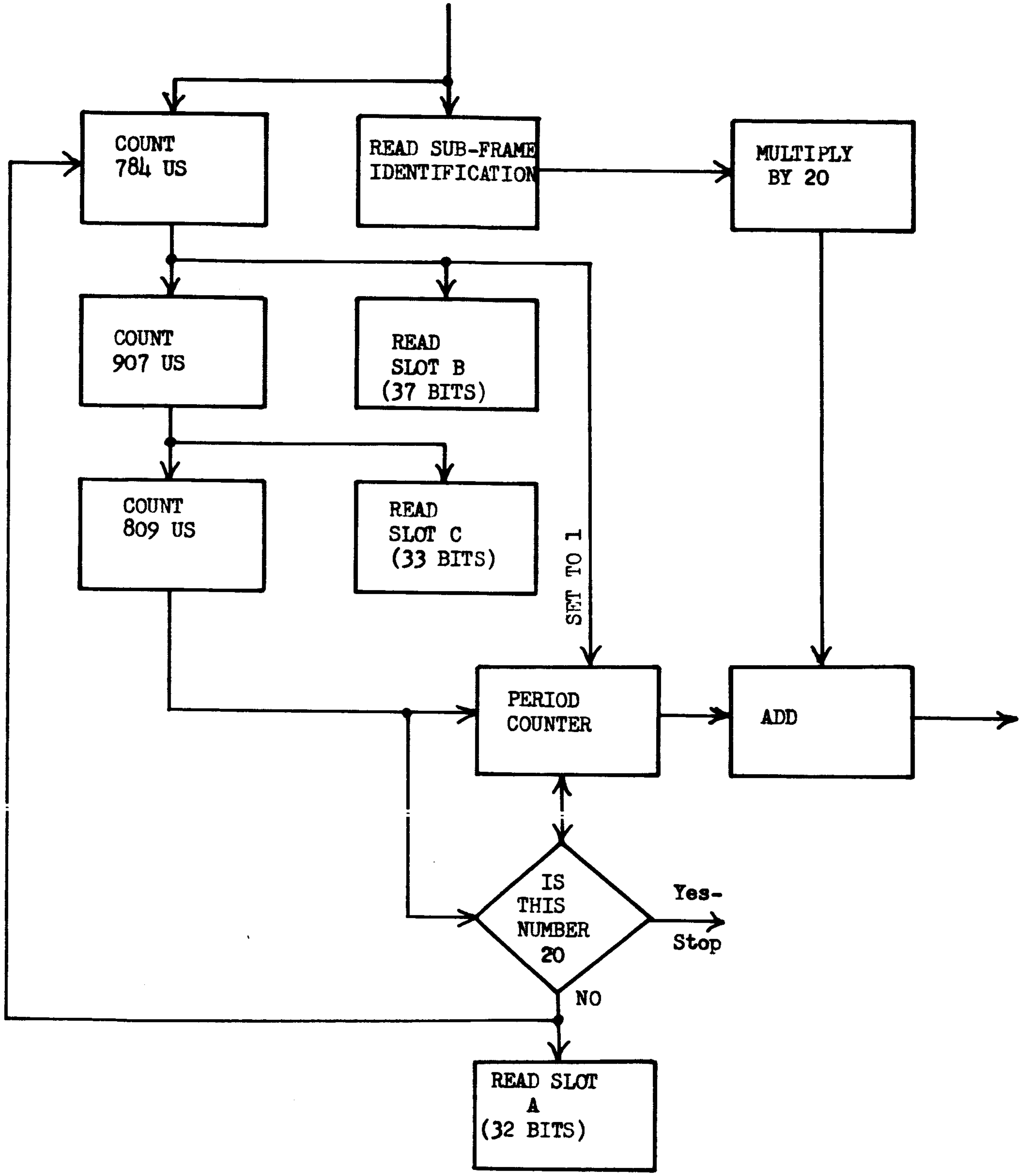


FIGURE 13. FLOW CHART OF FIRST STAGE DECOMMUTATOR

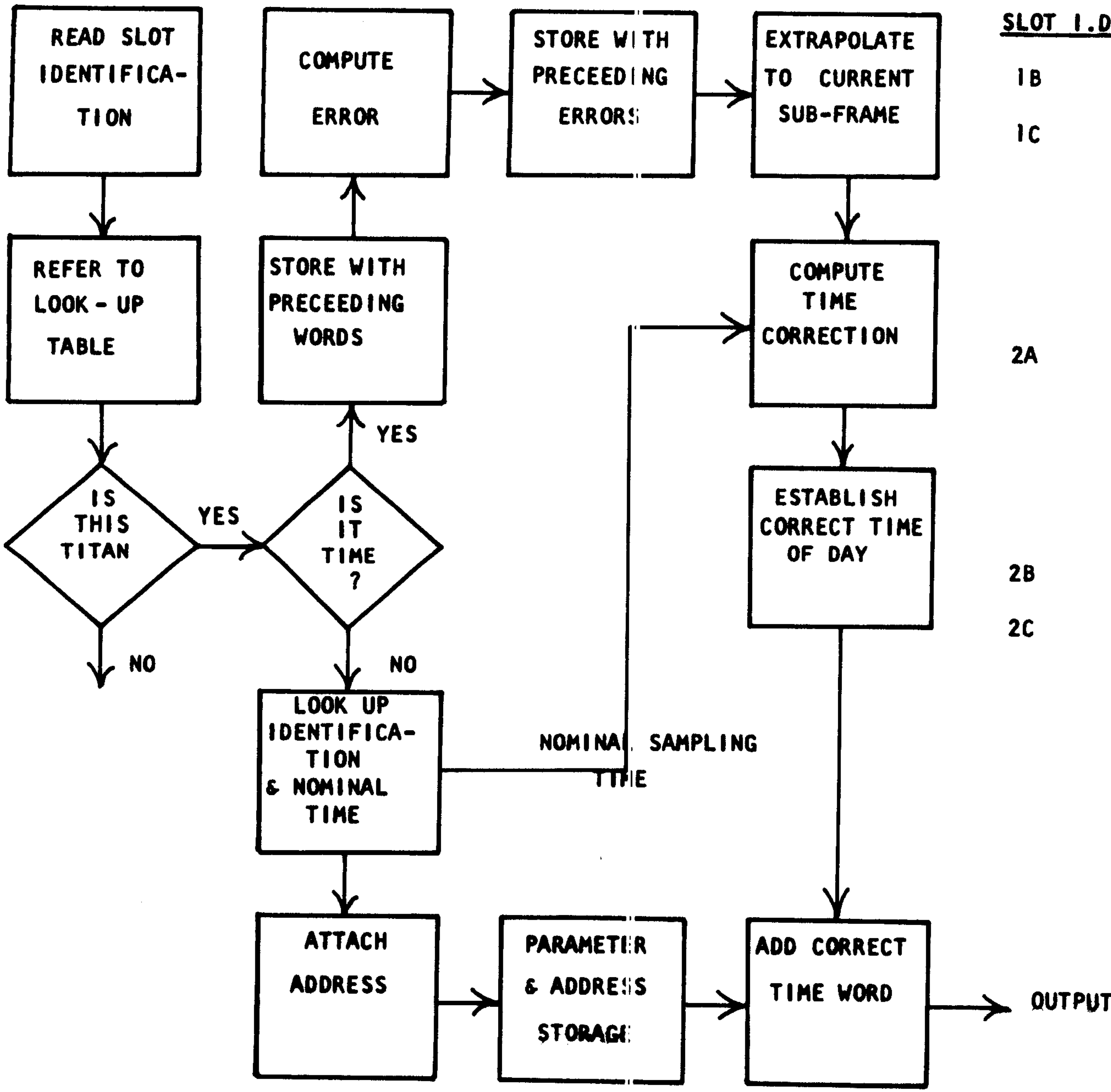
(2) The "B" slot of the initial period of each Burst sub-frame contains a 37 bit time of day word. This word represents the time that Titan sync was received and is in hours, minutes, seconds and microseconds. It occurs contiguously with slot "A" and is readout immediately following readout of the sub-frame identification.

(3) The "C" Slot of all periods is nominally the same length and contains 32 data bits plus one guard bit. Under conditions where the Titan input rate is in error from its nominal, the "C" slot in the last period of each Burst sub-frame may be lengthened or shortened a maximum of one (1) bit. The actual number of bits in these particular "C" slots are controlled by the Burst message and the decommutator must accept this condition.

(4) The first period of each Burst sub-frame is initiated by the occurrence of Burst sync, and the subsequent 19 periods are internally generated. The completion of each period in a sub-frame is counted by a period counter which at any time will contain a number between 1 and 20. Logic will cause another period to be initiated and slots A, B and C recycled every time slot "C" is readout and the period counter contains a number other than 20. The "A" slots in these intermediate periods are 32 bits long and the "B" slots are 37 bits long and these are readout accordingly. Every time Burst sync occurs, the decommutator begins reading the "A" slot in a Burst period. If the sync occurs at the nominal Titan sync rate, the last "C" slot of the previous sub-frame would have been completed, the period counter would have contained the number 20, and the logic would have prevented recycling of the next period. The occurrence of sync then resets the period counter to one (1) and, therefore, recycling will be allowed immediately after slot "C" is readout.

If the Titan sync rate is in error and sync arrives early, readout of the 5 bit subframe identification word begins before the guard bit in the last "C" slot has been completed. The guard slot was provided to allow this to happen and no conflict occurs because these two readouts are occurring simultaneously. If, however, the occurrence of sync were allowed to immediately reset the period counter, the number contained therein would be one (1) rather than 20 when the slot "C" readout was completed 25 microseconds later and recycling of intermediate periods would continue. This would result in erroneous outputs and cannot be allowed. To prevent this from happening, resetting the counter is delayed to occur simultaneously with the beginning of slot "B" readout. This assures that a counter content of 20 will exist at the time the slot "C" readout is completed and recycling will be inhibited.

(5) The output of the first-stage decommutator will be separate blocks from which parameters will be extracted in the second-stage decommutator. In order for this to be accomplished, the identification of the Burst slot from which the block was extracted must be conveyed to the second-stage decommutator. This is accomplished by forming output words



LOOK-UP TABLE

SLOT I.D.	SOURCE	CONTENTS	TIME
1B	TITAN	TIME	
1C	TITAN	177	44.9
		180	196.4
		190	1128.3
		196	1950.7
2A	TITAN	157	347.9
		161	1508
		169	3307.2
		176	4180.9
2B	GEMINI	TIME	
2C	TITAN	177	2544.9
		180	2696.4
		190	3783
		196	4450.7

FIGURE 14- SECOND STAGE DECOMMUTATION OF TITAN BLOCKS

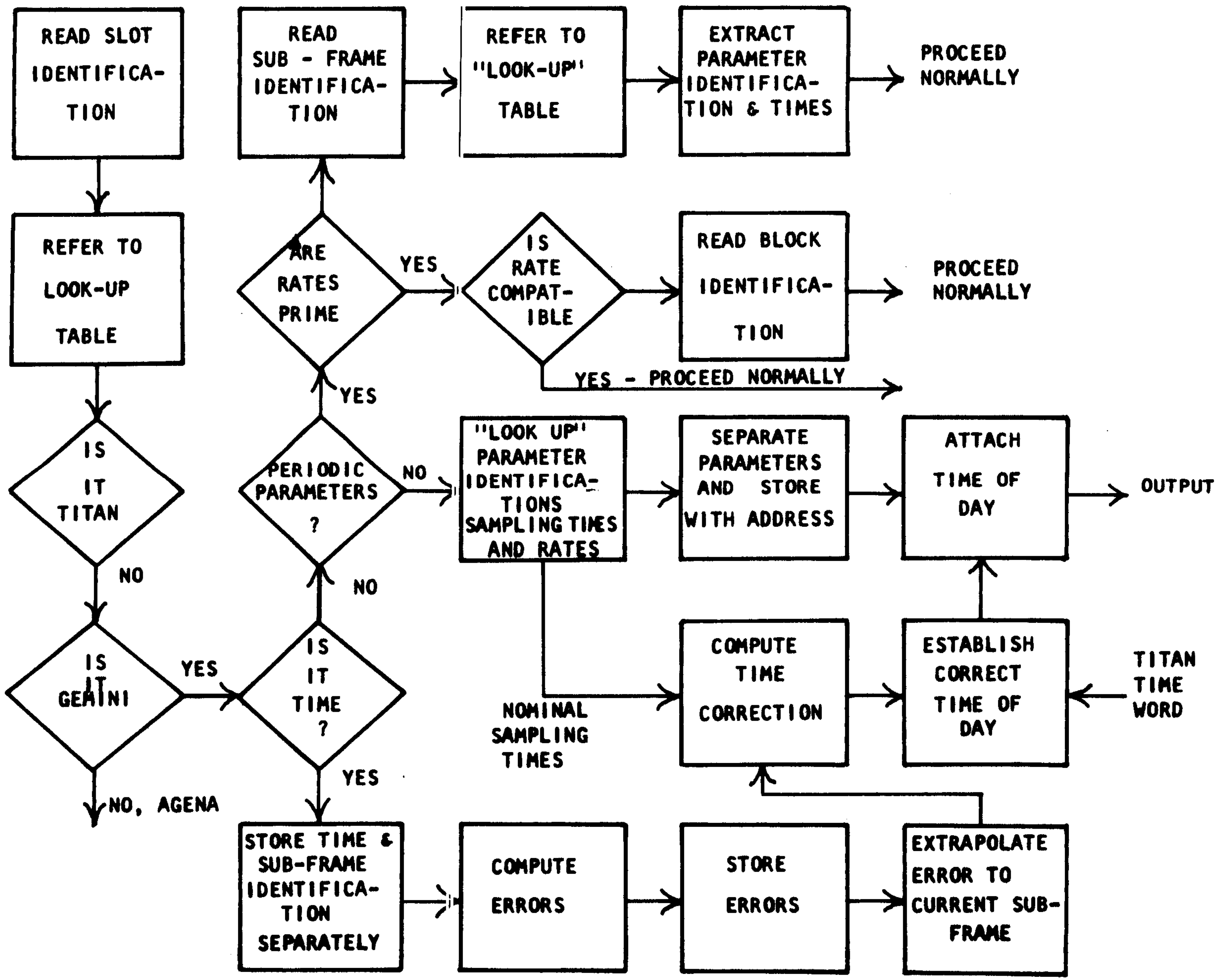


FIGURE 15 - DECOMMUTATION OF GEMINI BLOCKS

which consist of the slot identification as well as the data block.

The slot identification is developed by multiplying the sub-frame identification number (0-19) by 20 and adding the contents of the period counter. The channel identification (A, B or C) is obtained directly from the sequencer which controls the readout of individual blocks. The slot identification consists of 11 bits. Nine (9) of these identify a period between 1 and 400 and the other 2 identify slot A, B or C. The maximum data word is 37 bits long, hence, the first-stage decommutator output contains a data word of 37 bits. If the actual data word is less than 37 bits long (as is usually the case) the spare bits will be ignored.

2.7.2.2 Second Stage Decommutator

In this unit, individual parameters must be extracted from the blocks. These parameters must then be identified and the times at which they were sampled must be established. After that has been done, an output in a form suitable for recording, processing, or display must be assembled.

The blocks differ as to length, content, data rate, data periodicity, sampling rate (prime or sub-commutated), and whether a pseudo-synchronous or asynchronous relationship exists with the Burst message. The logic required to decommutate a block depends on the individual characteristics of a particular block. Hence, the second stage decommutator must be much more complex than is the first stage decommutator.

Figures 14 and 15 are flow charts which illustrate the various processes required of this unit which operates as follows:

(1) As each block is received, the slot identification number is read and pertinent information is extracted from a "look-up" table. The look-up table will have been pre-loaded with information which relates the telemetry format with the parameter selection and Burst format program. This table will contain all the information needed to establish the characteristics of a particular block in order that the proper decommutation process can be applied, and parameter identification and timing can be established. Table XVI illustrates the contents of a partial "look-up" table.

If it is found that a block contains Titan time, the word is stored in memory along with the preceding Titan time word. The two values are then subtracted and the difference is compared with the nominal Titan sync period (50,000 microseconds). Any discrepancies represent the time error which has accumulated during the previous sub-frame. The accumulated time error is shifted to a memory which contains time errors computed during previous sub-frames. From these values, the time error trend is calculated and extrapolated into the current sub-frame. This results in an estimate of the time error which will accumulate in the 50,000 microsecond interval after the latest time word.

Table XVI

Decommutation "Look-Up" Table

11	2	10	17	9	8	1
Slot 1.0	Source	Content	Time	Rate	Commutation/ S.F. Ident.	Periodicity
1 B	Titan	Time		20		Yes
1 C	Titan	177	44.9	400	Prime	Yes
		180	196.4			
		190	128.3			
		196	1950.7			
2 A	Gemini	Time		40		
		S.F.I.D.		40		
2 B	Titan	157	347.9	200	Prime	Yes
		161	1508			
		169	3307.2			
		176	4180.9			
2 C	Titan	177	2544.9	400	Prime	Yes
		180	2696.4			
		190	3783			
		196	4450.7			
3 A	Agena	Time		16		Yes
		S.F.I.D.				
3 B	Titan	121	241.3	100	Prime	Yes
		130	5708.2			
		133	6327.4			
		156	9482.4			

If it had been determined, from the look-up table, that the block contains Titan data, processing is as illustrated in figure 14, and the identity and nominal time of each parameter in that block is extracted from the look-up table. The block is then separated into individual parameters and the proper address is attached to each. The nominal time of each parameter is used to linearly proportion the estimated time error, thereby establishing a correction factor. The most recently stored time word is then read from memory and the nominal time of sampling plus correction is added. This results in a time word which defines in hours, minutes, seconds and microseconds, the time at which the particular parameter is sampled. The resulting time word is attached to the parameter and address and forms the output of the decommutator. One complete output word contains 37 bits of timing, 10 bits of address and 8 bits of data.

If the block had been found to contain Gemini time, it is known that the word consists of two parts. The first 7 bits of this word identifies the Gemini sub-frame and the remaining bits define the time of Gemini sync in seconds and microseconds. The "Time" portion of the block is extracted, transferred to the Gemini Time memory, and the Gemini Time error which will accumulate during the nominal Gemini period (25,000 microseconds) is estimated, in the same manner as Titan time error.

The Gemini sub-frame portion of the "Time" block is also stored in a memory from which it can be read whenever it is needed.

If the block had been found to contain Gemini data, the decommutation process would involve more steps than required for Titan blocks as shown in figure 15. A block of Gemini data could contain prime parameters transmitted periodically. If this were found to be the case, the decommunication process would be identical to the Titan process which was previously described.

A block of Gemini data could exist in which sub-commutated parameters are periodically transmitted. This would result in a block of different parameters being located in a particular slot during different Burst frames (section 2.6.2.2). When a block containing this type of data is received, the sub-frame identification number which had been received as part of the Gemini "time" block is read from memory and the look-up table is again consulted to determine the identity and time of the parameters in that particular slot during that particular sub-frame. After that has been done, the remainder of the decommutation process proceeds as for prime parameters.

In section 2.4.2.1.1 a synthesis process was described which allows parameters which occur at a rate which is not compatible with the 400 slot per second Burst message to be included. This process results in different parameters being included in the block which is transmitted in a particular slot during different Burst frames. It will, therefore, be necessary to uniquely identify each such block. Blocks of this type will be first transmitted in channel "B" which has a capacity of 37 bits, 32 of which contain data samples. The remaining 5 bit capacity is, therefore, available to handle a block address word and will be used for that purpose.

Hence, when a block of this type is identified, the block address will be read and the identity and time of each parameter will be obtained by again consulting the "look-up" table. After that, decommutation proceeds as before.

Another type of problem will exist if the block is found to contain parameters which are not periodically transmitted. Blocks of this type were described in section 2.5.2.3.3. With this type block it is necessary to determine how many separate parameters are involved. A memory cell is assigned to each parameter, and all samples of that parameter which were contained in the block are read-out in the sequence of occurrence and stored in the assigned memory. The identity of each parameter is obtained from the look-up table and attached to the memory cell. The nominal time that each sample occurred is also obtained from the look-up table and used to derive the current time of each sample. The correct time is then attached to each sample.

A local clock, operating at the nominal sampling rate of the parameters, will then readout words consisting of the parameter address and the time and value of each sample.

If, when the block was received, it had been found that it was from Agena, rather than Titan or Gemini, the processing would have occurred in the Agena channel. The procedure, however, would have been identical to that described for Gemini since the same types of blocks would be possible.

2.7.3 Equipment Characteristics

The assembly of blocks and the formation of the Periodic Burst message requires equipment to select and store data, to control the switching sequence in which the blocks are read-out, and to establish the time intervals for performing the switching. The extraction of parameters from the message and the re-establishment of time requires equipment, at the receiving terminal of the system, which perform similar functions. The major characteristics of these equipment are discussed in this section.

2.7.3.1 Block Formation

The selection of desired parameters and the assembly of blocks require that memory be provided. In section 2.6 it was shown that these memories must be of the automating input/output continuous selection type to provide the needed versatility and assurance that the proper time tag can be derived.

Figure16 is a functional block diagram of such a memory. Each continuous selection buffer contains two identically arranged memory cells (A and B), an input section and an output section, plus a switching unit which interchanges the input/output connections to the memories.

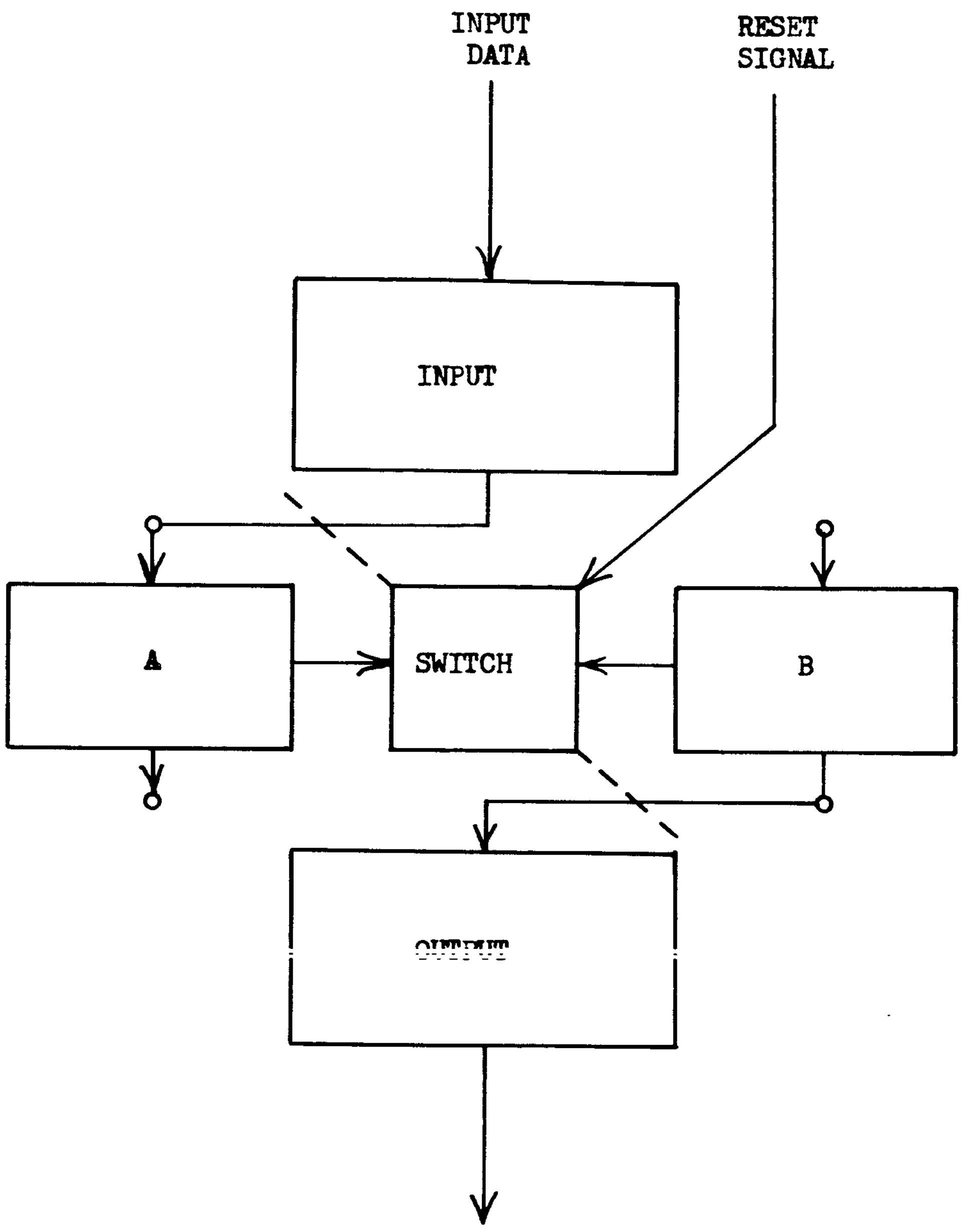


FIGURE 16. CONTINUOUS SELECTION STORAGE BUFFER

It has been assumed that all blocks will contain four 8 bit data words, hence, each storage section of a continuous selection buffer requires a 32 bit capacity. The total storage capacity per memory is, therefore, 64 bits. The input to the Burst system would be taken from the output of a standard decommutator. In order that desired parameters can be selected from the many which will appear at the input to the memory, it is necessary that an address memory section also be included in the continuous selection buffer.

The capacity of the address memory must be equal to 4 words times the number of address bits per word. The output of one type decommutator is in a format called Common Language, and contains a two part address of 11 bits each. One part identifies the input channel and the other part identifies the sub-channel. If the continuous selection buffer is to handle prime rate data only the channel portion of the address is important and the address memory would require a capacity of 44 bits (4x11). If sub-commutated parameters are to be selected the address memory would require 88 bits (4x22).

The address memory will be in the input sections, thus allowing a single address memory to serve both storage cells. The address memory will be programmable so that the identity of parameters to be selected can be changed as required.

The output section of the buffer contains the identification of that particular selection buffer. If a system were designed to handle 100 groups of parameters, the output buffer identification would require 7 bits. It would not be necessary for these identification words to be programmable.

The switching unit would be designed so that some specific event (occurrence of sync for example) would reset to switch to storage cell "A". When section "A" is filled by the occurrence of all selected words, a signal would be routed to the switch causing it to operate and interchange the input/output connection. After section "B" has been filled, the switch will again be flagged and the input/output connection will again be switched. In the meantime, memory section "A" would have been readout by action of the Burst message sequencer.

A continuous selection memory must be provided for each group of parameters which are to be assembled into a block. To assemble the blocks listed in section 2.5.3 for periodic Burst transmission requires the continuous selection buffers shown in Table XVII.

Table XVII

Data Rate	No. of Groups	No. of Selection Buffers
400	1	1
200	1	1
100	1	1
40	2	2
20	6	6
16	4	4
10	4	4
1.25	4	4
1.0	13	13
0.416	3	3
0.25	4	4
	Total	43

Each buffer requires 115 bits, hence, the total memory requirement is 4945 bits. From the above tabulation, it is readily seen that the same amount of memory is required to assemble a block of 400SPS parameters, as is required to assemble a block of 1SPS parameters. Hence, the total memory required increases as the number of blocks of low rate parameters increase.

It should be noted that in computing the memory requirements for the above table, a specific number of 20, 10, 1.25, 1, 0.25, 0.416 groups for the synthetic chamber of section 2.5.3(c) 4 and 5, were assumed. Also, no consideration was given to the number of subcommutated parameters in these groups, therefore, the actual memory size would have to be increased by 44 bits for each sub-commutated group.

All time words are readout before they reoccur and, therefore, continuous selection memories are not required. In addition, the Burst sync block contains a pattern which is internally generated plus a 5 bit Burst sub-frame identification address. The added memory required to handle all timing words and sub-frame address is 126 bits.

The total selection memory required is, therefore, 4945 bits + 44 bits/sub-commutated group.

2.7.3.2 Burst Message Sequencer Program

The function performed by the Sequencer is essentially that of a stepping switch. The Burst message contains 1200 slots, hence, this "switch" must contain 1200 positions.

Each "switch" position would be a programmable memory and the unit would be caused to sequentially step through these positions (memories) by externally generated commands. The word stored in each "position" would be the identification of the selection buffer which is to be readout at that time. Each word would contain 7 bits, thereby allowing the sequencer to address up to 127 difference sources. These sources could be any of the selection buffers previously discussed, the Burst sync generator, or the "time" storage units.

The sequencer memory requires a 1200 x 7 = 8400 bit capacity and must be programmable in order that the block readout sequence can be changed.

2.7.3.3 Timing Sub-Unit

The timing sub-unit generates the commands which cause the sequencer to step through its positions. The unit contains only 5 bits of memory but requires a precision frequency source (clock) and six high speed counters. The counters operate at a 1MC rate and contain various numbers of bits. These counters and their sizes are listed below:

Delay Counter	13 bits
"A" Slot Counter	10 bits
"B" Slot Counter	10 Bits
"C" Slot Counter	10 bits
Event Counter	5 bits
Limit Counter	5 bits

2.7.3.4 Burst Synchronizer

The logic described in section 2.2.2.3 for synchronizing the Burst decommutator would be instrumented with counters and memory circuits. The principle function of this logic is to establish the width of a search window and to position that window where subsequent sync patterns should occur. Whenever sync is received the unit turns off the sync comparison circuit, to prevent the possibility of an accidental recognition, until a time interval has expired and the sync pattern should re-occur. In this particular case, where sync occurs every 50,000 $\pm$ 25 microseconds, the turn-off interval must be 49,975 microseconds and the window must then be opened for 50 microseconds. If sync is not again received, the turn-off time must be reduced by 25 microseconds to 49,950 and the width of the window must be increased to 100 microseconds.

This control can be obtained by a "Sync Delay Counter" operating in conjunction with a "Window Width Counter". The "Sync Delay Counter" would have to count a maximum of 49,950 microseconds and at a 1MC rate would require 16 bits. The width of the search window increases by 50 microseconds each time an expected sync is not obtained and the total width would depend on the number of "no sync" indications allowed. This has not been established but is estimated to be no more than 10. Hence, the maximum required count in 1000 microseconds and 9 bits are needed.

Every occurrence of sync is stored in memory and every occurrence of "no sync" is stored in a different memory. It is estimated that the most occurrences of either which must be stored would be 10, hence each of these memory would require 4 bits. In addition, the synchronizer requires a sync pattern memory of about 28 bits.

2.7.3.5 First Stage Decommutator

The function of this unit is to extract individual blocks from the Burst message, identify the slot position in which the block had been received, and provide an output word for each block which contains the received data plus address. The Burst message consists of a serial train of contiguous blocks which occur at intervals controlled by the timing sub-unit.

In order to decommutate this message it is necessary that this unit contain equipment for timing the beginning and end of each slot, and for uniquely identifying each slot. These functions can be achieved by counters and memories. The 5 bits which occur immediately after sync is a binary number between 0 and 19 which identifies the Burst sub-frame which will follow that particular sync pattern. These 5 bits are read-out serially as they occur and are routed to a "multiply by 20" circuit, the output of which is transferred to an "add" circuit where it is stored in memory. The number which is to be stored varies between 0 and 380, and, therefore, a 9 bit memory is required.

Immediately after the last sub-frame identification bit is read, two events occur simultaneously. The period counter is reset to one (1) and the first "B" slot of the sub-frame begins to be read. This particular slot contains the "Time of Titan Sync" word in hours, minutes, seconds and microseconds, and is 37 bits long. The capacity of the communication circuit limits the speed of bits to 1 bit per 24.5 microseconds, hence 907 microseconds are required to read this word. While this block is being read, the content of the period counter (1 in this case since it was just reset) is added to the output of the "multiply by 20" circuit, and the period count (1 to 400) is obtained at the output of the "add" circuit.

After the "time word" has been received, the Burst period number and the slot identification are attached and the entire word, consisting of 8 bit period I.D., a 2 bit slot I.D. and 37 bits of timing is shifted to the output. These 48 bits must be read-out after the complete time word has been received and while the following slot is being read in. Hence, the "time slot B" must contain 37 bits of memory. The 48 bit output word must be read-out in less than 785 microseconds (time required to receive the next 32 bits at 24.5 microseconds per bit).

Figure13 is a flow chart of the First Stage Decommutator. As can be seen, this unit contains a precision clock, count circuits, and memories as well as logic and arithmetic circuits. The following table lists the quantities and capacity of the various equipment used in this unit.

Precision Clock	1	$1x10^{-8}$
Slot Time Counter	3	10 bits
Period Counter	1	5 bits
Slot Memories	1	37 bits
	2	33 bits
Period Counter		9 bits

2.7.3.6 Second Stage Decommutator

This unit extracts individual parameters from the blocks, identifies each parameter, and reconstructs the sampling time of each. The flow chart for this unit is presented in figures 14 and 15 and its operation is described in section 2.7.2.2. The characteristics of the various memory and arithmetic sections are defined below:

2.7.3.6.1 Look-Up Table - - 135,000 bits

The Look-Up Table is a memory which contains all information necessary to identify particular parameters in a block, establish the nominal time of each, and provide decommutation circuits with data which will control the further processing of each block. The size of that memory is computed as follows:

(a) Slot Identification - 12,980 bits

When a block is received for decommutation, it will be accompanied by an address word which identifies which of 400 period and which of 3 slots (A,B or C) that block was extracted from. A total of 1180 slots must be identified (20 were used for Burst sync) and each requires 11 bits. Hence, 12,980 bits are needed.

(b) Source Identification - 2360 bits

Blocks are assembled from one of three links (Titan, Gemini or Agena). The particular link must be identified for each of 1180 slots, and 2 bits are required. The total is, therefore, 2360 bits.

(c) Periodicity Identification - 3540 bits

Certain blocks may contain parameters which are not periodic. These must be decommutated by a unique method and must be identified. It is also necessary to know the nominal rate of all periodic parameters. In this particular case, only two such type blocks would exist (640 or 160SPS). It would, therefore, be necessary to provide 2 bits of memory which could be used to indicate that the parameters are periodic or are non-periodic at either rate A(640) or B(160). Hence, the total memory is 3 x 1180.

(d) Commutation Identification - 2360 bits

Parameters contained in a block will be at Prime, super-commutated or sub-commutated rates. In addition, parameters may exist which were programmed by synthesis and which are not compatible with the 400 slot per Burst channels. These have characteristics which resemble sub-commutated data. The look-up table is organized so that super-commutated and prime parameters can be handled in the same way. Hence, these need not be separately identified. Sub-commutated, and synthesized blocks which resemble sub-commutation, must be handled in different ways and must be identified. Hence, each of the 1180 blocks must be identified as to one of the three types and 2360 bits are required.

2.7.3.6.3 Decommutation of Prime Rate Parameters

When a block of prime rate parameters occur, the individual values are shifted into separate registers in the proper channel (Titan, Gemini or Agena). The parameter identification is also shifted into the same register. The nominal time of sampling words will be transferred into a memory in the "time correction" circuit. That unit will then proportion the predicted "error" and obtain a corrected time estimate. The unit will then read-in the time word from storage, add the "correction" and transfer the "correct" time word to the register which contains the address and value of the parameter. The entire word is then shifted out. The memory required to accomplish this per parameter is:

	Titan	Gemini	Agena
Parameter Values	8 bits	8 bits	8 bits
Parameter Address	8 bits	8 bits	8 bits
Temporary Time Storage	16 bits	15 bits	17 bits
Computed Time Correction	5 bits	4 bits	6 bits
"Correct" Time Word	37 bits	37 bits*	37 bits*
Total	64 bits	62 bits	64 bits

Each block can contain up to 4 parameters, hence, four complete registers must be provided for a total of 64x4 = 256 bits for Titan, 62x4 = 248 bits for Gemini and 64x4 = 256 bits for Agena.

2.7.3.6.4 Decommutation of Non-Periodic Parameters - 282

A block of this type data may contain 1, 2 or 4 samples which may have been obtained from either 1 or 2 different parameters. The number of samples in such a block may vary from one occurrence to another, however, the four registers must be provided to handle the maximum case and the system must operate whether all registers are loaded or not. When a block of this type occurs, each sample is shifted into a register along with the identification of the parameter associated with that sample. Nominal time of sampling is shifted to the error corrector and a correction is obtained as for prime rate data. Each individual sample is treated as a separate parameter except that the same parameter address may be attached to more than one sample. After time corrections are available, they are shifted to the proper register where the entire word is held in storage. The amount of memory required up to this point is the same as for a Gemini prime rate block (although the procedure has been somewhat different) and is 256 bits.

To read the word out, however, requires that an internal clock be automatically set to the proper data rate, and the clock then read-out the proper register. To instrument this would require logic in the clock control circuit to contain 2 bits to set the clock rate and 12 bits to enable it to consecutively address any one of the 4 registers. As each register accepts a data word, the clock would be flagged and the identification of that register would be inserted into clock address unit in the order in which that particular register is to be read.

* The Hour and Minutes portion of this word is derived from Titan time.

Three bits additional are, therefore, required in each of the registers. The total memory requirement is:

Parameters with Address and Correct Time		256 bits
Clock Rate Control		2 bits
Register Identification		12 bits
Register Address		12 bits
	Total	282 bits

2.7.3.6.5 Decommutation of Sub-Commutated Parameters

The contents of a block of sub-commutated parameters appearing in a particular slot will vary depending on the sub-frame from which the parameter was selected. It is, therefore, necessary to establish which sub-frame is involved before the parameters can be identified. To facilitate this, the information contained in the "look-up table" for all slots which contain sub-commutated parameters, will be sub-catagorized by sub-frame identification. Hence, when a sub-commutated block is received, the identity and nominal times of parameters is still obtained from table.

Each Gemini (and Agena) "Time of Sync" word contains a sub-frame identification number. When a block of sub-commutated parameters is received, the sub-frame I.D. is read and the parameter identification is obtained from the "table". The value of each parameter and its identification is then shifted into registers. In order that time reconstruction logic can be simplified and the length of time words, both in the "look-up table" memory and the time "correction" logic, can be kept to a reasonable length, two restrictions are imposed on the assembly of blocks of sub-commutated parameters. First, all parameters in a block will occur during a single sub-frame, and the maximum time word is limited to a sync period or 17 bits. Second, all sub-commutated blocks will be read-out one sub-frame late; thus allowing a full choice of parameters to be included in every block and establishing the reference time for each parameter to be that of the previous sync word, which is still being held in memory. The correction of time is then accomplished as for prime rate data. The correct time word is then generated by read-out the next preceding time of sync word and adding the correction.

The memory required is:

		Gemini	Agena
Sub-Frame Identification		7 bits	7 bits
Output Words		256 bits	248 bits
	Total	263 bits	255 bits

2.7.3.6.6 Decommutation of "Synthesized" Rates

The synthesis procedure, described in section 2.4.2, when used to format data from rates which are not compatible with a 100 slot Burst channel, results in a pseudo-subcommutation rate. Blocks of this data will appear in different slots of different Burst messages and each block must be uniquely identified. This is accomplished by including a block identification word with each block of parameters. This block I.D. is read and the corresponding parameter identification and times are extracted from the "look-up table". The rest of the process is identical to that used for prime rate parameters. The total memory requirement is:

	Gemini	Titan	Agena
Output Word Assembly	248	256	256
Block Identification	5	5	5
Total	253 bits	261 bits	253 bits

2.7.3.6.7 Summation of Second Stage Decommutator Memory Requirements

Look-Up Table		135,000 bits
Titan Time		145
Gemini Time		123
Agena Time		147
Prime Rate Decommutation	Titan	256
	Gemini	248
	Agena	248
Non-Periodic Data Decommutator		282
Decom. of Sub-Com. Data	Gemini	263
	Agena	255
Decom. of "Synthesized" Data		
	Titan	261
	Gemini	253
	Agena	253
	Total	137,734 bits

2.7.3.7 Tabulation of Equipment Requirements

		Total Memory	Counters Quantity	Counters Size
(a)	Block Formation	5000 bits		
(b)	Message Sequencer Program	8400 bits		
(c)	Timing Sub-Unit		1	13 bits
			3	10 bits
			2	5 bits
(d)	Burst Synchronizer	36 bits	1	16 bits
			1	9 bits
(e)	First Stage Decommutator	145 bits	3	10 bits
			1	5 bits
(f)	Second Stage Decommutator	137,734 bits		
	Total	151,315 bits	5	- 5 bits
			1	- 9 bits
			6	- 10 bits
			1	- 13 bits
			1	- 16 bits

2.8 Extrapolation to a General Application

The criteria developed in previous sections, for implementing a Periodic Burst System, is summarized below:

2.8.1 Characteristic of the Burst Format

(a) A Burst period equals the period of the highest input data rate that is to be retransmitted periodically.

(b) All Burst periods must be divided into the same number of slots so that the total number of slots in the Burst message exceeds the total number of input blocks (sum of all input samples rate).

(c) All slots of a particular Burst channel must be of the same length.

(d) All slots are identified by period number and Burst channel as 1A, 5B, etc.

(e) When multiple asynchronous inputs are available, one of them must be selected and used to control sync of the Burst message.

(f) The Burst message must be correlated with the input link sync at a rate higher than the Burst message rate, usually the input link sync rate. The actual rate at which this correlation is required depends on the maximum possible input rabe error as pecified for the telemetry link, and the maximum time error which will be allowed to accumulate. If the accuracy of the specified input link rate and the rate of link sync are so low that the allowable error will be exceeded in the time between successive correlations, one of the other links should be selected to control the Burst message.

(g) Time of occurrence of link sync must be transmitted in a slot of the Burst message. By comparing the difference between successive time of sync with the nominal (expected) difference in time the actual error in input rate can be estimated. That error can be linearly proportioned to each data parameter in the Burst message in accordance with the nominal time of occurrence of that parameter, thereby correcting the time associated with that parameter.

(h) The period format must be adjusted to provide an unused "guard band" at its end to prevent loss of data when the input rate is above nominal.

2.8.2 Assignment of Blocks in the Format

(a) Successive blocks will appear in slots which are separated by the ratio of "total slots per second" to "input data rate".

(b) Slot separations which are not related to the total number of slots by an interger, will not appear in the same slots of successive Burst messages.

(c) In order that slots which will be used by blocks of data having non-compatible separation in subsequent Burst frames can be fully used in every frame, it is necessary that these slots be identified and blocks from separate input channels assigned in a unique order. These slots can be identified by synthesizing and formatting an input rate which is related to the actual non-compatible input rate by an integer and which also results in a slot separation which is related to the total slots available as an integer

(d) The initial block of data from all input channels must be placed in a slot in one of the periods between the beginning of the Burst message and the period which corresponds to the separation of subsequent blocks of data at that rate.

(e) Slots in the Burst format must be assigned for the transmission of "time of sync" from each of the asynchronous links.

(f) These slots must be periodic at a rate corresponding to the sync rate of each link.

(g) The initial sync word from each link must be placed in a slot within one of the Burst periods which occur between the beginning of the Burst message and the period corresponding to the "time of sync word" rate of each link.

(h) One Burst channel will consist of blocks from an input channel which has the same data rate as the Burst Period rate.

(i) A second Burst channel will contain blocks from various combinations of the next higher compatible input rates.

(j) Compatible rates are those which are related to each other as an integer and which have "slot separation" which are related to the total number of slots in the Burst channel as an integer.

(k) Compatible "inputs" can often be synthesized from non-compatible inputs. For this to be accomplished, it is necessary that the rate of the synthesized "input" be related to both the Burst rate and the actual input rates as an integer, and it must also result in slot separations which are related to the total slots in the Burst channel as an integer.

(l) A Third Burst channel would contain time of sync words for the link to which Burst is correlated, and for the other links which provide inputs. It also contains blocks from other compatible input channels, as well as blocks from synthesized "inputs".

(m) Input rates could occur such that synthesis would not be possible. In that case, blocks from those inputs could be non-periodically transmitted.

2.8.3 Implementation of the Periodic Burst Technique for Arbitrary Conditions

In the previous sections of this report, a technique was defined and a systems criteria established for the periodic Burst retransmission of data from asynchronous Titan, Gemini and Agena PCM telemetry links via a 40.8KBPS communications circuit. In this section, those techniques and criteria are extrapolated to cover the general case where the input links and the retransmission circuit are arbitrarily selected. It is assumed that the number of input links, the characteristics of each, and the capability of the communications circuit, are defined prior to an operation and, therefore, this section is an outline of the procedure for designing a format under arbritrarily given conditions and for programming the system. Limitations which exist in each step of the procedure are defined.

2.8.3.1 Establishing Burst Periods

The technique defined in the previous section requires the retransmission bandwidth to be divided into a number of periods and each period to be divided into a number of slots. Each slot contains space for a block of data to be periodically transmitted.

Figure 5 illustrates the basic features of the Periodic Burst Format. It can be seen that the message (Retransmission Bandwidth in KBPS) is divided into "N" periods per second, and each period is further divided into "M" slots and a guard band. The design of the Burst Message begins by determining values of "N" and "M".

To define the Burst period, it is necessary that each input link be examined and the following characteristics of each extracted:

(a) Input data rates from each link.
(b) Number of parameters at each rate.
(c) Synchronization rate of each link.
(d) Stability of each link.
(e) Number of bits per data word.

Figure17 is a logic Flow Chart of the process involved in defining the number of periods to be included in the Burst format. The procedure is described below:

Step 1: Compute lowest number of periods which will be compatible with all input rates. In order for data at any given rate to be periodically transmitted, it is necessary that the rate be related to the number of periods in the format by an integer. Hence, when more than one rate is to be transmitted, the number of periods must be selected so that integer relationships exist with all rates. That is, number of samples (N_1) at rate No. 1 is to rate number 1 (R_1) as the number of samples (N_2) at rate No. 2 is to rate number 2 ($R2$) etc. Mathematically:

$$\frac{N1}{R1} = \frac{N2}{R2} = \frac{N3}{R3} \text{ etc.} = K$$

where "K" is the number of periods required in the output format for all input rates to be periodically transmitted. This value of "K" is the preferred choice for period rate and is used in the first attempt to design the format.

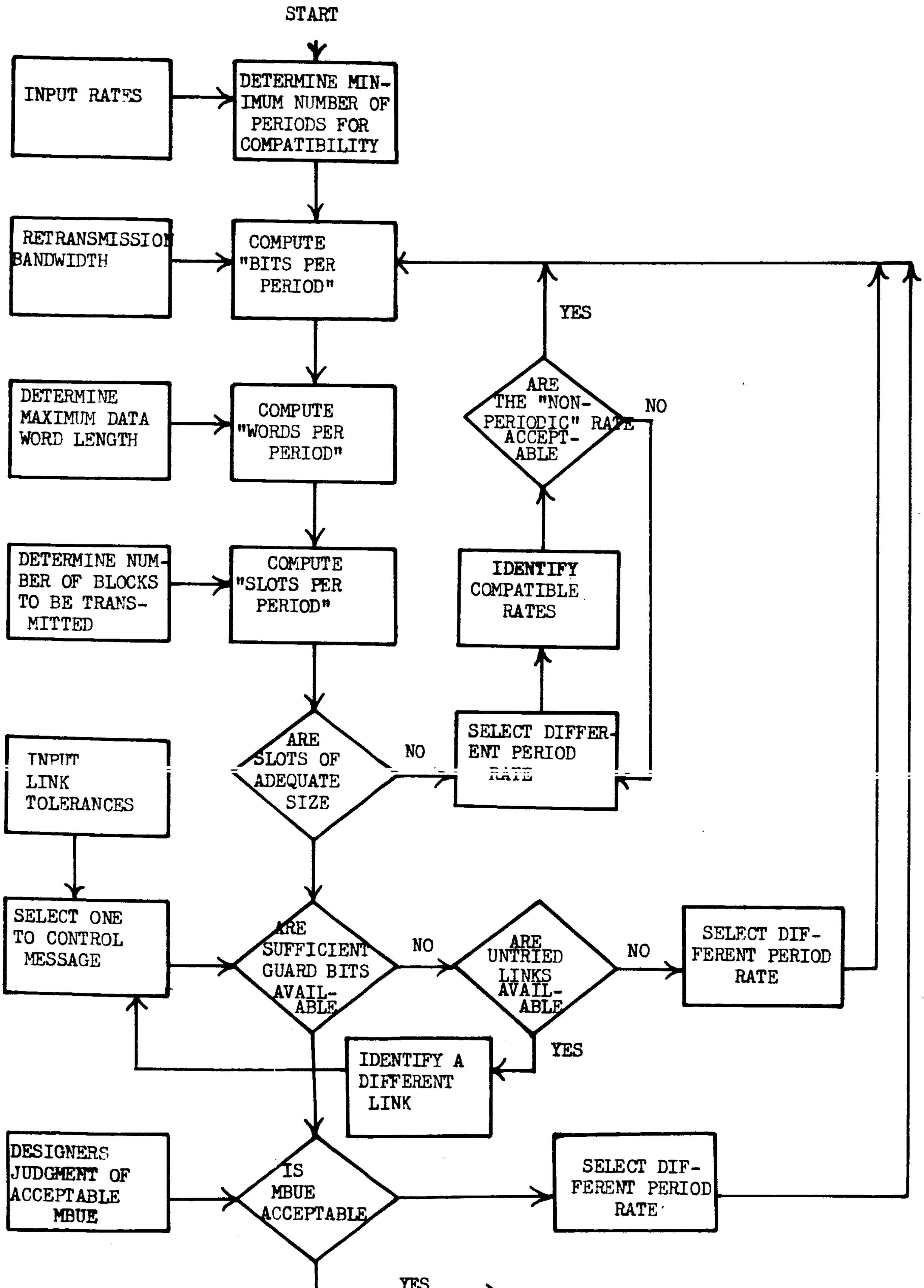


FIG. 17 FLOW CHART ILLUSTRATING THE DESIGN OF A PERIODIC BURST MESSAGE

Step 2: Compute Bits per Period. The number of bits which can be retransmitted during a period depends on the retransmission bandwidth (B) in bits per second and the period rate (k) in periods per second. Hence, "P" (period) $= \frac{B}{K}$.

Step 3: Compute Words per Period. The number of words which can be transmitted in an established period depends on the length of each word. Information concerning the length of all data words is extracted from the input link characteristics. If all data words are the same length, that length is noted and used to compute the words per period. If different length words are involved, the longest is selected as the basis of the calculations. This assures that all length words can be transmitted, albeit with a reduction in bandwidth utilization efficiency whenever the shorter words are transmitted. After word length has been established, the computation of words per period involves dividing "P" (step 2) by the word length.

Step 4: Compute Slots per Period. In the Periodic Burst system a data block will contain an integer number of parameters at a given rate from a particular link. In order that these blocks can be transmitted, each period must be divided into slots where the blocks can be placed. In a general case, the number of blocks will exceed the number of periods, and each period must contain more than one block. In order to simplify the future design of the sequencer which controls the transmission of blocks, it is required that (except for minor exceptions to be discussed later) all periods be divided into the same number of slots. Dividing the total number of blocks to be transmitted per second by the number of periods per second will result in the minimum number of slots which must be provided in each period. Because fractions of a block cannot be efficiently transmitted, it is necessary that each period be divided into an integer number of slots. Hence, if the computation of slots per period, obtained by dividing periods per second by blocks per second, results in a mixed number, that value must be increased to the next higher integer. For example, if it is computed that 2.3 slots are needed per period, it will be necessary to provide three slots per period.

Step 5: Determining Adequency of Slot Sizes: After the number of slots which must be provided in each period has been determined, a basis exists for testing the suitability of the design to this point. First, it is obvious that for the required number of blocks to be included in a period with each block containing some integer number of words the total period must be at least as long as the number of bits per word times blocks per period times words per block. It then follows that the number of words per block must be established. This is done by again considering the operational requirements. By determining the total number of words which must be transmitted each second, and dividing that number by the number of slots in the message, the average number of words per block can be computed. It is not required that all blocks within a single period be of equal size, however, correspondingly placed blocks in all periods must be the same size. Furthermore, the number of words in a period must equal the average words per block multiplied by slots per period. If this product is not an integer, the number must be increased to the next higher integer. For example, if the average word per block is 2.75 and each period contains 3 blocks, the total bits per period must equal 3x2.75-8.25 raised to 9, the next higher integer.

If the period is large enough to handle the required number of bits, a logic "yes" is obtained and the design of the message can continue. If, however, a logic "no" is obtained the size of each period must be increased. To make a change in the period size, it is necessary that a new period rate be selected. Now it should be remembered that the initial period rate was the lowest rate compatible with all input rates. Hence, when this period rate is reduced, some input rates will no longer be compatible and therefore cannot be periodically Burst transmitted.

In selecting the new lower period rate, it is necessary to identify which of the input rates will be periodically transmitted and which of them it would be acceptable to transmit non-periodically. After this has been accomplished, a new period rate which is compatible with the identified "periodic rate" is computed. This new period rate is then fed back into the logic at step 2, and steps 2, 3, 4 and 5 are repeated. After a "yes" is obtained at step 5, the design continues.

Step 6. Determine if sufficient guard bits are available. The sync rate and frequency tolerance of each link is extracted from the input link characteristics. One link is selected to control the formation of the Burst message. The maximum time error which could occur between sync words on the selected link if the frequency was at the maximum specified error is calculated. This time is then converted into bits on the retransmission circuit.

The number of bits correponding to that error is added to the number of bits per period containing data words and compared with the number of bits which can be transmitted in a period. If the period cannot handle the required data plus guard bits, a logic "no" is obtained from step 6. When the logic "no" is obtained, a check is made of the sync rates and tolerances of the other input links to determine if one of these would require fewer guard bits. If such is found to be the case, this other link is selected and again the availability of guard bits is tested. If a link cannot be found which requires few enough guard bits to satisfy the period, it is necessary to select a lower period rate. The lower rate is tested for compatibility with the rates to be periodically retransmitted and, if compatible, is fed back into the logic at step 2. All intermediate steps are then repeated until a logic "yes" appears at the output of step 6.

Step 7. Determine if maximum bandwidth utilization efficiency is acceptable. At the completion of Step 6, the characteristics of the Burst period which can handle the required data is available. This remaining test is to determine if, by using the defined period, a satisfactory utilization of the retransmission bandwidth can be achieved. A definition of what is acceptable depends on many things, such as the priority of parameters which can be included in the designed periods and the priority of the parameters which were not included, the availability of additional communications circuits, the cost of establishing or using other circuits, etc. In the final analysis, however, the definition of acceptable MBUE is a matter of the judgement of the designer. It should be noted, however, that some slots will be required for Sync, Timing and possibly parity, and the actual bandwidth utilization efficiency will be lower than the MBUE calculated by dividing bits of data per period by total bits per period. The authors of this report believe that the MBUE should be at least 90% and a goal of 95% should be sought. If an acceptable MBUE is not achieved, a new period rate which meets the compatibility criteria should be selected and the design logic repeated from Step 2. After a satisfactory MBUE has been achieved, the periods and, therefore, the basis for the message format has been defined.

2.8.3.2 Assignment of Blocks

In the procedure outlined in Section 2.8.3.1, Burst periods and slots were defined. Also, a decision was made as to which link would control the Burst message, and input rates which must be non-periodically transmitted were defined. The next part of the process is to establish specific parameters to be assembled into each block, and to define the slot location in which each particular block will be transmitted. This is accomplished as follows:

Step 1. Examine the formats of each input link and extract. and tabulate. the sampling rates and number of parameters at each rate. This will result in a compilation similar to Table I.

Step 2. Determine the channel number of all parameters at each rate and decide which parameters are to be Burst transmitted.

Step 3. Select all parameters to be assembled in separate blocks, and assign a particular continuous selection buffer to assemble the block. Note the address of the buffers which have been selected for each block.

Step 4. Assign Burst sync to slot 1-A, and to subsequent slots which are separated by the ratio of total slots per Burst channel to the sync rate of the channel which was previously selected to control the message. Load the address of the Burst sync generator into proper positions of the sequencer.

Step 5. Assign "Time of Link Sync" to Slot 1B, if the periods are divided into three or more slots, or to slot 2A if the periods are divided into only two slots. Load the address of the time memory in the proper position of the sequencer.

Step 6. Assign a block of the highest periodic rate parameters to slot 1C. Subsequent blocks of the same parameters will appear in the "C" slot of every period. Load the address of the buffer which contains these parameters into every "C" slot of the Sequencer.

Step 7. Determine suitable slots for the still unassigned blocks. Begin with the highest rate parameters and make assignments in consecutive order to the lowest rate. If more than three slots per period are available, assign the D, E, etc. slot first. After these are fully assigned, proceed with assignments in Burst channel A, leaving channel B to still be assigned. Note the addresses of each buffer and load those addresses in the proper positions in the Sequencer.

Step 8. Assign "Time of Sync" blocks to "B" slots of suitably located periods. Load the addresses of each "Time of Sync" memory into the proper positions of the Sequencer.

Step 9. Assign remaining blocks to "B" slots and load the addresses in corresponding positions of the Sequencer. "Non-compatible" rates which are assigned by synthesis should always be assigned to "B" slots so that space can be available for including a block identification word.

2.8.3.3 Adjustment of Timing Sub-Unit

Figure 12 is the Block Diagram of the Timing Sub-Unit. This part of the procedure is to adjust the various counters in this unit so that the Sequencer will be caused to address the various memories at the proper time.

Step 1. Adjust "Sync Delay" so that the delay is greater than the control link tolerance multiplied by the control link nominal sync period.

Step 2. Adjust "A" counter to the time required to transmit a block in burst channel "A".

Step 3. Adjust "B" counter to the time required to transmit a block in burst channel "B".

Step 4. Adjust "C" counter to the time needed to read a "C" block plus a guard band.

Step 5. Adjust "Period Counter" to the number of periods per sub-frame.

Step 6. Adjust "Limit Counter" to be equal to the guard band included in step 4.

2.9 Periodic Burst Simulation Program

This program simulates a technique by which decommutated telemetry data is accepted from three independently transmitting sources, sampled, and selected data is reformatted and sequentially retransmitted preserving the original data sample rates.

2.9.1 General Description

The principal part of the program, which simulates real time operational capability of the retransmission processor, is written in FAP. Two FØRTRAN subroutines, labelled SLIP and CARDS, are used for reading into memory the conditional information specified for a particular operating period. Subroutine SLIP reads in the time variance, if any, specified for Titan data. Subroutine CARDS reads in the selected data channel identification numbers, source codes, and the associated data sampling rates. The FØRTRAN subroutines read in these data from the data deck in a one-time operation. The main program first calls the FØRTRAN subroutines and then initializes itself for real time simulation using the specified time variance value and data channel selections.

The real time simulation consists of reading data from the four input tapes in the sequence of decommutation, determining if the data is a sync word or selected channel data, and reformatting the data accordingly for output records. One (normal) output tape is written which records the serial retransmission format of the selected data, plus sync time words and sync count for each source sync word received, and includes a master sync word for each output record. A second (auxiliary) output tape is written which records all decommutation times and data word inputs from the four input tapes in the sequence of processing.

The output tapes are processed off-line to obtain hard copy printouts which can be correlated to similar hardcopy printouts of the records on the input tapes. The sequencing of data and times on the output tapes is interlaced in such order as to enable reconstruction of all actual processing times to the accuracy of linear interpolation.

2.9.2 Input Parameter Cards

The data deck for the simulation program consists of the following:

Card 1 - Time Variance Card (Format I4)
Card 2 - Selected Channel Count (Format I4)
Cards 3 to N - Channel Specification Cards (Format 2I4, E8.3)

Time variance is specified in units of 1 microsecond per data channel decommutation at the highest sampling rate, namely the Titan 400-Sps. To represent the worst case situation of accumulated time of processing slippage the program attributes the effect entirely in one direction to Titan data times only during each resync period of the processor, nominally 50,000 microseconds.

The selected channel count may be any integer up to a maximum of 122 which specifies the actual number of channel identity cards which follow.

The channel specification cards give the source code (1 to 3) and channel identity number in the first two fields (2I4) respectively, and the pointed-decimal value of the data sample rate in the third field (E8.3) in units of 1 sample per second, to an accuracy of three decimal fraction places. The number of channels selectable for each combination of a data source and sample rate is limited by the maxima specified in Table XIX, which must not be exceeded. If the data deck contains too many channel identity cards for a particular source and sample rate subroutine CARDS and/or the main program will malfunction and come to an error halt before accepting any input data from tapes.

Table XVIII Number of Channels (Maximum)
Selectable for Source and Rate
(Source Code 1 = Titan, 2 = Gemini, 3 = Agena)

Source Code	Rate	No. of Chans
1	400.	4
1	200.	4
1	100.	4
1	40.	8
1	20.	20
2	40.	8
2	10.	2
2	1.25	8
2	.416	24
3	16.	16
3	1.	4
3	.2	20

2.9.3 Input Data Tapes

Data inputs to the system are simulated by four binary record tapes. One tape (RC 491) represents time data. The other three tapes represent the data outputs from the Titan (RC 1618), Gemini (RC 1619), and Agena (RC 1620) decommutators. The records on these input tapes are read into memory buffers on a reload needed basis. Individual data words from each source and the time words are processed by the simulation program in the sequence which represents actual data receipt and decommutation.

2.9.4 Output Tapes

Two binary output tapes are written by the simulation program. One (normal output) tape records the 60-time slot contents of burst message reformatted data representing the serial retransmission output of the telemetry burst processor. The second tape represents all data and channel identities read from the input tapes, decommutated and interlaced with the associated times of decommutation. If the time variance specified by data card 1 is other than blank or zero, the time of decommutation of Titan data is cumulatively modified by the program to represent time slippage within each resync interval of the processor.

2.9.5 Hardcopy Printout

The two binary output tapes are printed out by the standard IBTAPE conversion routine of the FØRTRAN monitor system. The printout represents four 36-bit data words to a line. Each data word is printed in 12-digit octal code. Every odd order data word is decimally numbered. The data words belonging to each output record are grouped together and each output record is headed by a sequential decimal number.

2.9.5.1 Burst Message Records

The first output record is the initial sync record and represents only sync information up to the time of the first Titan sync. Word one contains the master sync pattern and sync count equal to 1. Word two contains the time, modified for slippage, that the first Titan sync was detected. Word(s) 7 and/or 37 contain the latest (up to 2) times that Gemini sync was detected and the corresponding Gemini sync count. Word 4 contains the latest time, if any, that an Agena sync was detected and its sync count number. All other words contain zeros since retransmission positions for data cannot be fixed until after master sync conditions are established, based on the first Titan sync. All other output records are in normal 60-word formats. Words 1, 2, 7, and 37 contain the master sync and sync times, as for the initial sync record. The Agena sync time word commutates through the sequence: word 4, word 19, word 34, word 49, absent every fifth message, and then repeats. All normal format 36-bit data words represent a single time slot of reformatted data, i.e., up to four 8-bit data words at a single sample rate. They are unpacked by dividing the 12-digit octal code into four 3-digit octal data words. The time slot assignments are given in Table XX.

Table XIX Burst Retransmission Time Slot/Data Assignments

Source Code	Data Sample Rate	Time Slot Assignments	(1)Max No. Data Chans
1	400.	3,6,9,12,15,...,60	4
1	200.	5,11,17,...,59	4
1	100.	8,20,32,44,56	4
1	40.(First group)	13 & 43	4
1	40.(2nd group)	28 & 58	4 (2)
1	20.(First group)	25,40,46,55	16
1	20.(2nd group)	14,26,38,50	16
2	40.(First group)	22 & 52	4
2	40.(2nd group)	28 & 58	4 (2)
2	10.	16 & 31 (Odd msgs)	8
2	1.25	31 (Even msgs)	8
2	.416	16 (Even msgs)	24
3	16.	4,19,34,49	16 (3)
3	1.	10	20
3	.2	None	0

Notes:

(1) The max no. of data channels in TableXVII reflects the no. which could be accommodated in the time slots of the periodic burst retransmission format. This does not necessarily mean that that many channels at that rate are actually available in the independent source periodic burst formats. It could be more or less. TableXVIII reflects the lesser maximum.

(2) Time slots 28 and 58 are shared by Titan and/or Gemini 40-Sps ~~data on a basis of not more than 4 data channels assignable~~ for both sources.

(3) The 16-Sps Agena data channels are placed in the time slots 4, 19, 34, 49 interlacing the Agena sync time words between which they were decommutated.

2.9.5.2 Time-Channel Output Records

The time-channel output records all consist of thirty-eight 36-bit words. Words 1, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33 and 36 are all time words. Words 1, 9 and 15 represent times for the Gemini data following. Word 1 also represents time for the Agena data following. Words 1, 9 and 15 are not modified for time slippage effect. All other time words associate with the Titan data following and are modified for time slippage effect if any was specified. All other words represent data channel identity numbers and the data itself, if any, as decommutated. The initial 6 octal digits are always zero in the data channel words. The next 3 octal digits give the channel identity. The final 3 octal digits give the 8-bit data word or repeat the channel ID number to indicate absence of data. Words 2 and 3 give Agena channels in the order decommutated. Words 4, 5, 7, 8, 10 and 11 give Gemini channels in the order decommutated.

The remaining words give Titan channels in the order decommutated.

2.9.6 Simulation Program Operation

The periodic burst simulation program is executed as a job processing run under control of the FØRTRAN 2 system and monitor. The object deck consisting of binary program deck plus data deck and control cards is loaded from the card reader, or preferably from tape, after going card-to-tape on peripheral equipment. The system uses the following tape transport assignments:

Tape Unit	Assignment
A1	FØRT 2 (SYSLB1)
A2	CARDS IN
A4B	Periodic Burst Message Output
A5B	Time-Channel Record Output
B3B	Gemini Data Input (RC1619)
B4B	Time Data Input (RC491)
B5B	Agena Data Input (RC1620)
B7B	Titan Data Input (RC1618)

2.9.7 Simulation Program Logic

The simulation program logic and sequence of operation is given in the form of a simplified flow chart in Figure 19 The headings, such as IA1C1, which appear over the functional blocks in the flow chart, are the same as the headings used for the associated program segment in a complete listing of the program.

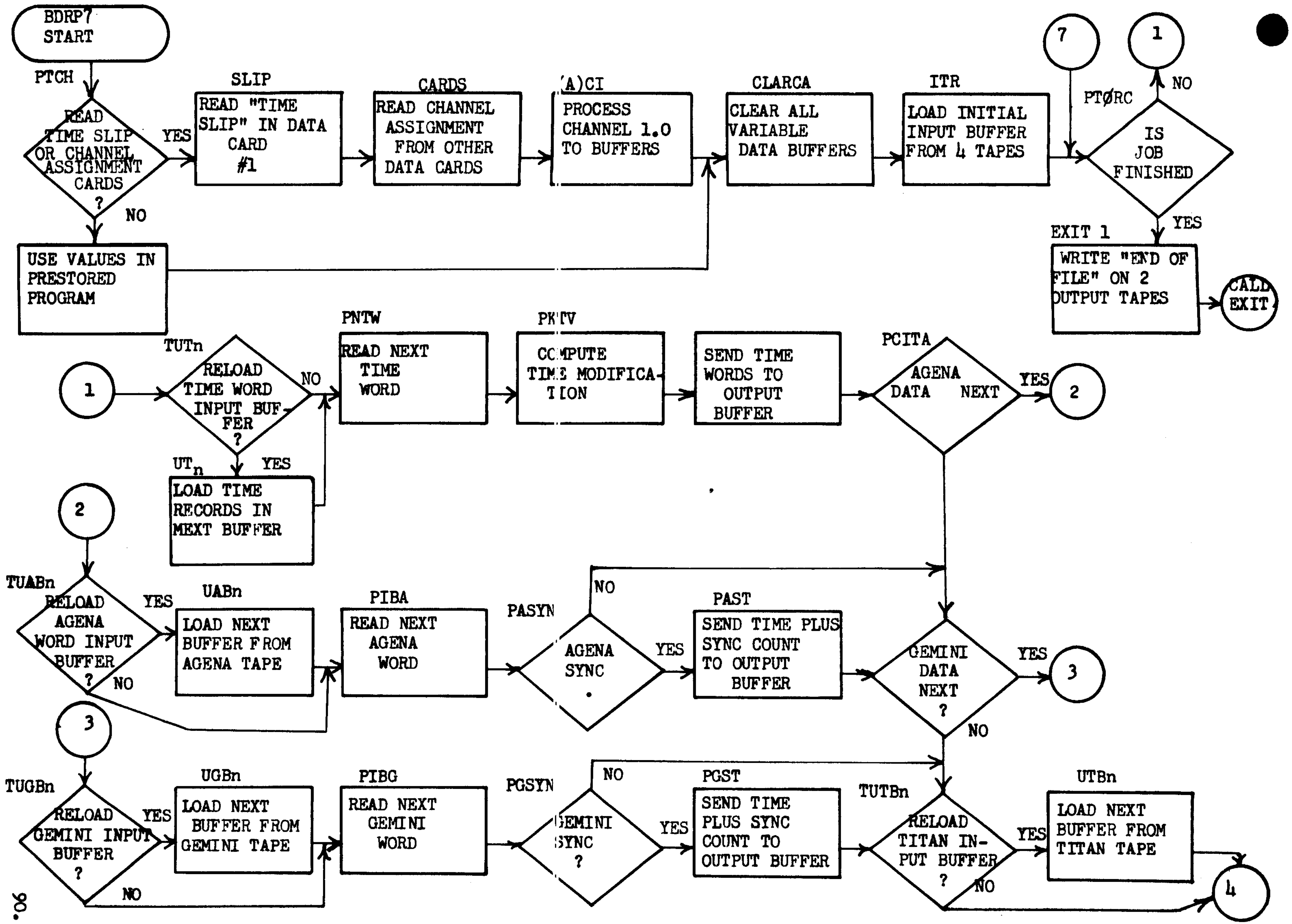


FIG. 18 SIMPLIFIED FLOW CHART, PERIODIC BURST SIMULATION PROGRAM (Page 1 of 2)

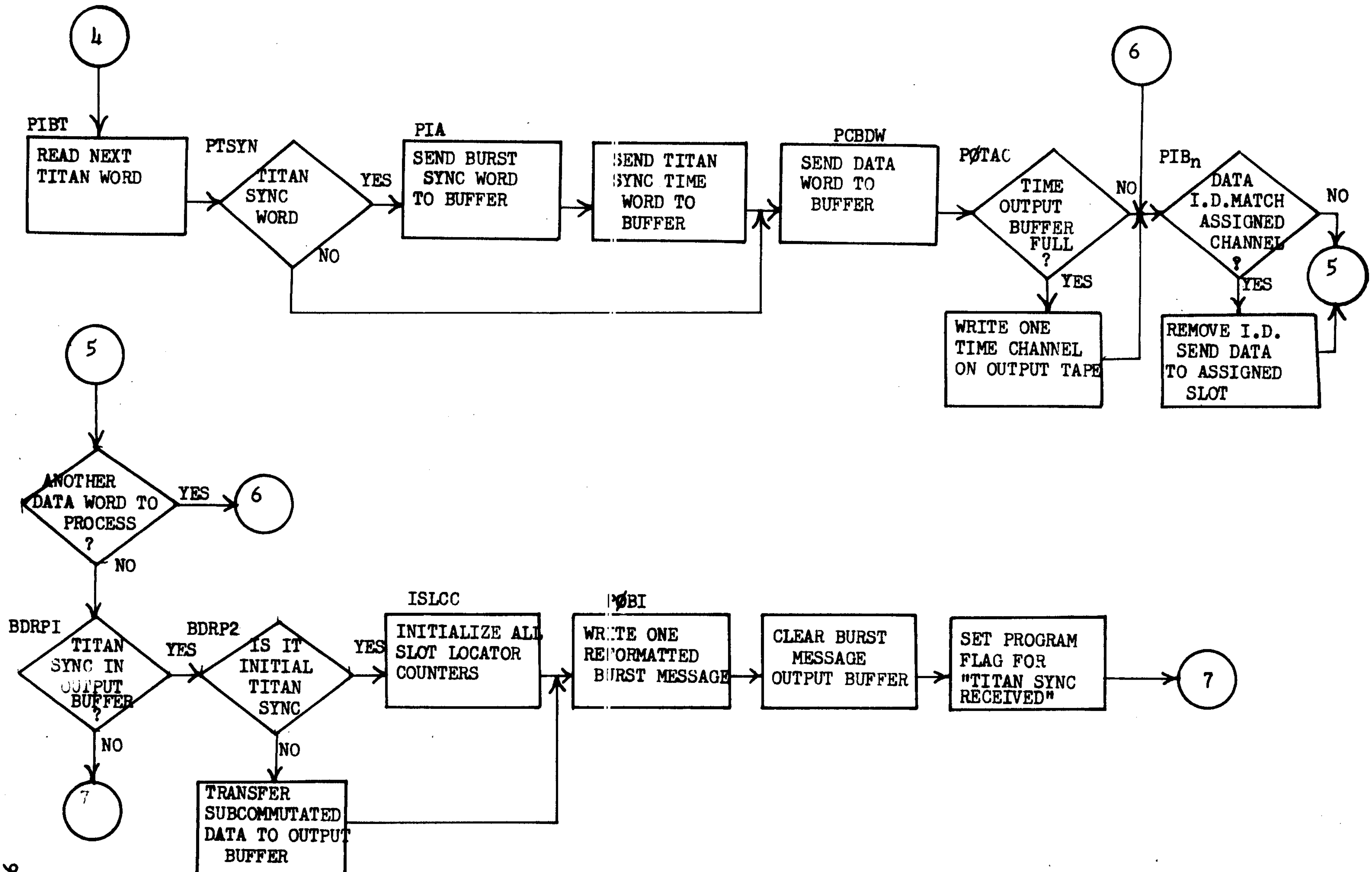


FIG. 18 SIMPLIFIED FLOW CHART, PERIODIC BURST SIMULATION PROGRAM (Page 2 of 2)

2.10 Demonstration

For the demonstration two different sets of parameters were selected to be "transmitted" in the simulated program. One of these sets contained 29 parameters preselected by RCA. A preliminary run of those parameters was made several days before the scheduled demonstration, and a print-out was available for examination and was discussed with the Technical Representative of the Government before the demonstration. The parameters (channels) contained in this set are shown in Table XX.

At the completion of the discussion, the Technical Representative was allowed to select a set of different parameters to be Burst "transmitted" during the demonstration. He selected 48 parameters which were assigned slots as shown in Table XXI. During the subsequent demonstration, this particular set of parameters could not be transmitted and the witnessed Burst message contained the preselected parameters which had been previously discussed. A later investigation as to the reason why the set of parameters in Table XXI did not work was made. This showed that the Technical Representative had been allowed to select 6 Agena 1.0SPS parameters, whereas the simulation program could only handle 4 such parameters. Two of the specified 1.0SPS Agena parameters (numbers 204 and 205) were deleted, thereby reducing the total number of parameters in the set to 46. The modified set was then run and a copy of the print-out is being forwarded to the Technical Representative.

Most parameters in the demonstration contained fixed values, numerically equal to the channel number. A few of them, however, contained actual data. One of the 400SPS channels selected by the Technical Representative contains a ECG record. That parameter was selected for analysis. Figure 19 is a plot of a portion of the ECG Data. It can be seen that the plotted portions of this data change rapidly.

For the demonstration, a delay equal to 53 microseconds per Burst period was inserted so that subsequent parameters actually occurred in 2500 microsecond intervals. The first 13 of these samples are plotted on an expanded time scale at the actual input rate of one sample per 2553 microseconds and is represented by the solid curve of Figure 20. The dashed curve of the same figure was constructed from readouts of the parameters at 2500 microsecond intervals where the time assigned to each parameter corresponded to the readout time. Extremely large errors resulted as shown

The "time of Titan sync" word, which appears as the second word of each Burst sub-frame (Slot B) was read for successive Burst sub-frames. A difference of 26,880 units was found. Each unit is equivalent to 1.9 microseconds and a time difference of 51,072 microsecond was found to exist. This difference was divided by the 20 periods per sub-frame and showed that data was occurring at 2553.6 microsecond intervals. From the design of the Burst message, it is known that blocks are readout at 2500 microsecond intervals.

Table XX

Preselected Parameters

Slot Assignments	Source Code	Rate	Chan ID Nos.					Total
"C"=Words 3,6,9,...,60	T	400.	181	190				2
"B"=Words 5,11,17,.,59	T	200.	169					1
"B"=Words 8,20,32,.,56	T	100.	131	149				2
"A"=Words 13 & 43	T	40.	95	98				2
"A"=Word 55	T	20.	15	23	51	71		4
"A"=Words 22 & 52(First 4)	G	40.	5	35	65	95	135	5
"A"=Words 28 & 58(5th)								
"A"=Word 16(Odd # Msgs)	G	10.	169					1
"A"=Word 31(Even # Msgs)**	G	1.25	163	164	165	166		4
"A"=Word 16(Even # Msgs)**	G	.416	173	176	179	182		4
"A"=Words 4,19,34,49*	A	16.	23	53				2
"A"=Word 10**	A	1.	129	144				2
								29

Notes: * In commutation sequence.

**As occurring in subcom sequence.

Table XXI

Parameters Selected by Technical Representative

Slot Assignments	Source Code	Rate	Chan ID Nos.								Customer Req'd	Subsequently Demonstrated
"C"=Words 3,6,9,....60	T	400.	190	191	200	201					4	4
"B"=Words 5,11,17...50	T	200.	159	169							2	2
"B"=Words 8,20,32,..56	T	100.	122	131	140	149					4	4
"A"=Words 13 & 43	I	40.	89	92	104	107					4	4
"A"=Word 55	T	20.	3	9	21	27					4	4
"A"=Words 22 & 52	G	40.	16	17	18	19					4	4
"A"=Word 16(Odd # Msgs)	G	10.	169								1	1
"A"=Word 31(Even # Msgs)**	G	1.25	161	162	163	164	165	166	167	168	8	8
"A"=Word 16(Even # Msgs)**	G	.416	170	171	172	173	174				5	5
"A"=Words 4,19,34,49*	A	16.	95	96	97	98	99	100			6	6
"A"=Word 10**	A	1.	200	201	202	203	(204)	(205)			(6)	4
											(48)	46

Notes: * In commutation sequence.

** As occurring in subcom sequence.

More than four Agena chans at 1-SPS cannot be handled by existing program. Causes overflow into next correlation buffer. Refult: program malfunctions and bumps itself off-line. Reason - Channel ID No. in this case = 204, is interpreted as the total buffer size and causes suicide by zeroing out the 198(=204-6) locations below that in which Chan ID No. 205 is placed.

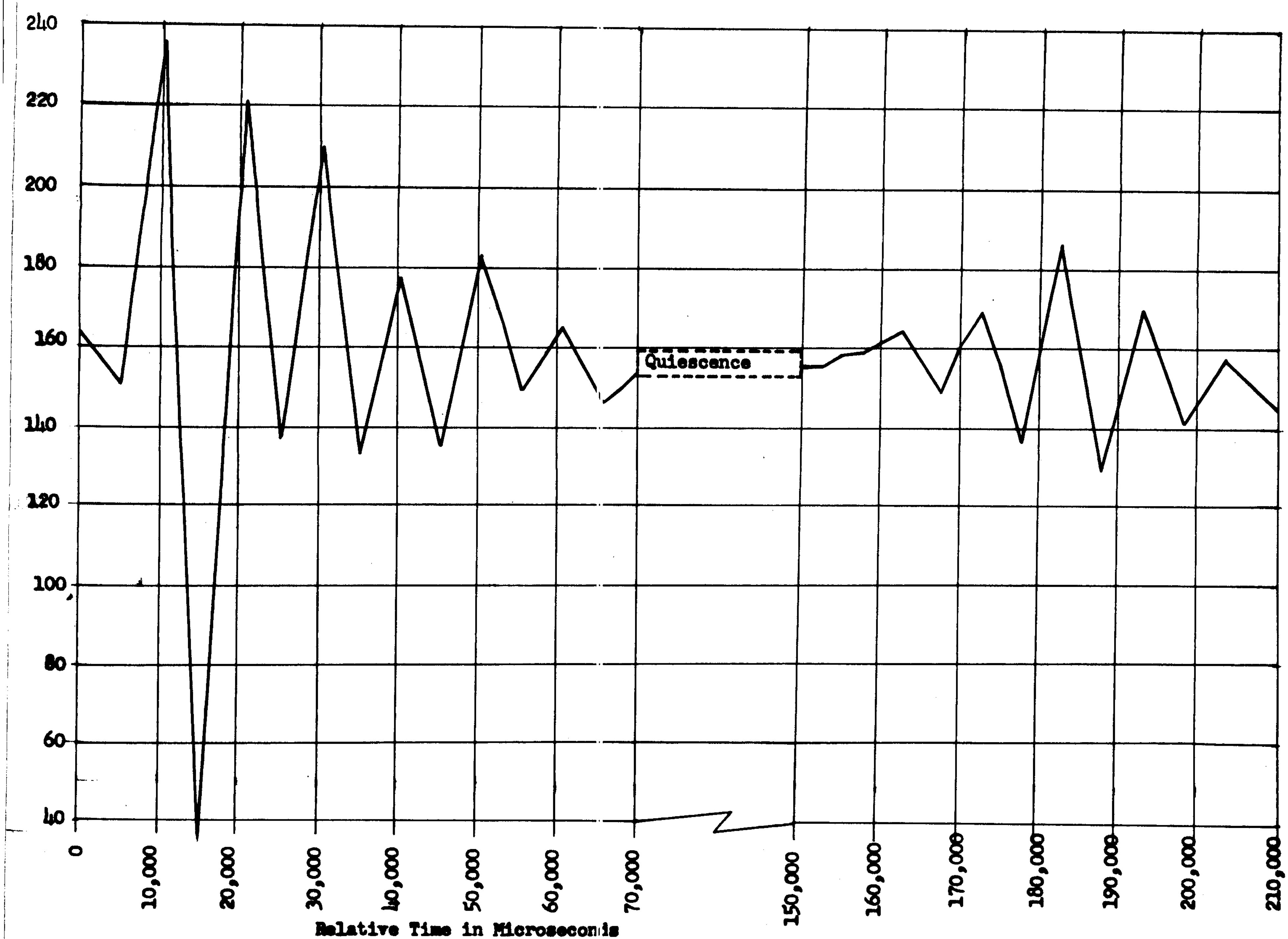


FIG. 19 ECG INPUT TO SIMULATED BURST SYSTEM

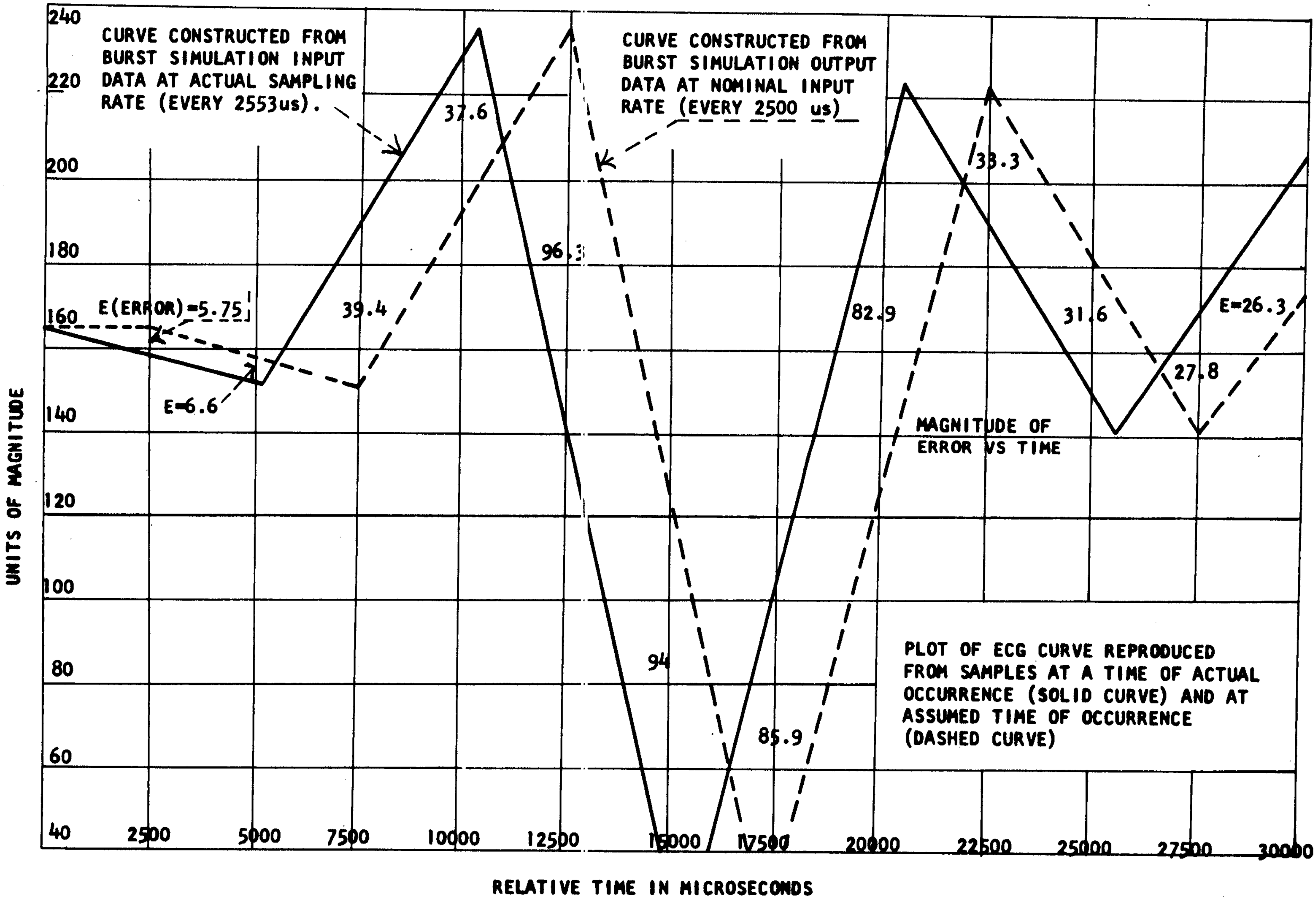


FIGURE 20 - COMPARISON OF ORIGINAL CURVE AND RECONSTRUCTED CURVE WITH GROSS TIME ERRORS

The occurrence of the first block is locked to the occurrence of sync, and will actually take place a fixed delay later. It is assumed that Burst readout will be delayed 1200 microseconds and will occur every 2500 microseconds thereafter. The time of the first sample is then established by subtracting the fixed delay from the readout time. These are identified on Figure 23 as R_1, R_2, R_3, etc.

If that fixed delay is subtracted from each block time a curve can be reconstructed. The dashed line in Figure 21 represents such a reconstructed curve. The errors at each readout time are shown. These errors represent the difference between the value of the parameter which was stored in memory 1200 microseconds previously and the value which should have been stored.

Table XXII lists the times of various samples and corresponding resampling, and the absolute magnitude of the resulting error. Several things can be noted:

First, the delay between readin and readout is continually decreasing; second, the time between the actual occurrence and the assigned time continually increases. Third, the absolute magnitude of error depends on the slope of the curve at the place sampled and the total time which has elapsed since the beginning of the sub-frame.

By determining the actual time between samples to be 2553.6 microseconds, the time of each sample can be more accurately established. This can be done by:

(1) Determining the difference between the input interval and the Burst interval, i.e., 2553.6 - 2500 = 53.6 microseconds.

(2) Reduce the fixed delay between readin and readout (1200 microseconds) in this case by 53.6 microseconds per block except the first. If this technique were assigned to sample number mix in Table XXIII, it would be found that:

   (a) The event occurred at t 12765
   (b) Readout occurred at t13700
   (c) The delay has decreased from 1200 by 5(53.6) or 268.
   (d) The assigned time should be 13700-1200+268=12,768 microseconds.

The 3 microsecond difference (12768-12765) is probably due to the 1.9 microsecond per unit factor, which was multiplied by the difference between successive Titan time words, being rounded off and slightly in error. In any event, on the curves of Figure 23, it is not possible to plot 3 microsecond, and the reconstructed curve would lie on the solid line. Hence, the error at Point 6 would be insignificant. Table XXIII is the residual error at each point plotted on Figure 23.

2.11 Conclusions and Recommendations

The Periodic Burst Technique has been found to provide a means for transmitting large amounts of PCM data from multiple asynchronous telemetry links, without introducing large asynchronous resampling errors. This capability is achieved by a unique way that time words are merged into the Burst format. An efficient utilization is made of the communication circuits bandwidth ($\approx$90%), the periodicity of individual samples is maintained, and the delay between the initial reception and the final display or processing at a user location of each parameter is less than one period at the rate of that particular parameter.

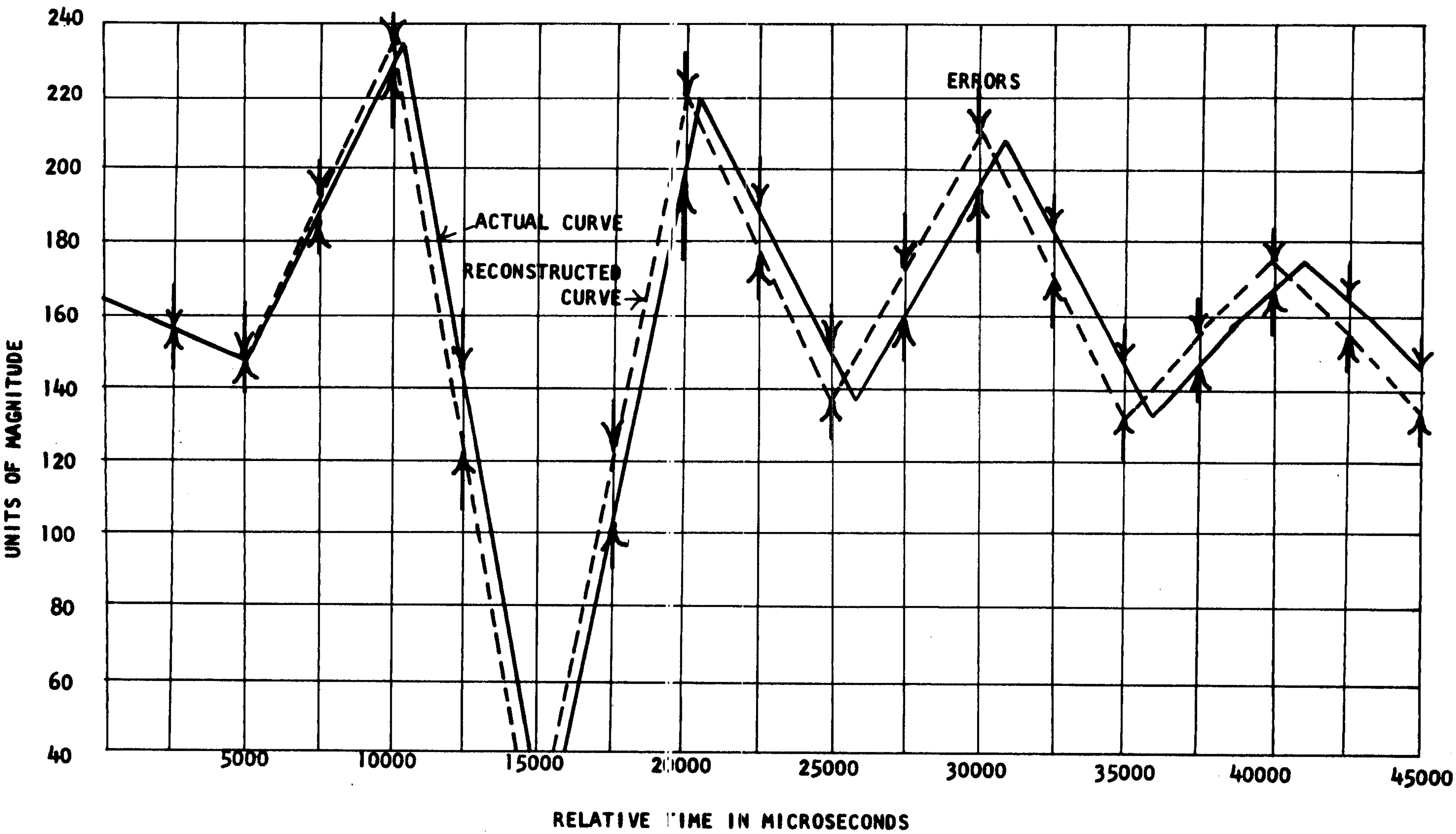


FIGURE 21 - ORIGINAL AND RECONSTRUCTED CURVES - AFTER PRELIMINARY CORRECTION

Table XXII

Magnitude of Error Resulting from Time Errors

Sample No.	Time Occurred	Time Read	Time Assigned	Absolute Magnitude of Error
1	t_o	t_{1200}	t_o	0
2	$t\ 2553$	t_{2700}	t_{2500}	1.0
3	$t\ 5106$	t_{6300}	t_{5000}	1
4	$t\ 7659$	t_{8700}	t_{7500}	3
5	t_{10112}	t_{11200}	t_{10000}	3
6	t_{12765}	t_{13700}	t_{12500}	10
7	t_{15318}	t_{16200}	t_{15000}	11
8	t_{17871}	t_{18700}	t_{17500}	12

Table XXIII

Residual Errors After Correction

Sampling Time	Time Error	Slope	Magnitude Error
0	0	-0.0026	0
2500	.6	-0.0026	-0.0016
5000	1.2	-0.0026	-0.0031
7500	1.8	0.017	0.031
10000	2.4	0.017	0.041
12500	3.0	-0.044	-0.128
15000	3.6	-0.044	-0.158
17500	4.2	0.0401	0.168
20000	4.8	0.0401	0.193
22500	5.4	-0.0166	-0.09
25000	6.0	-0.0166	-0.10
27500	6.4	0.0146	0.096
30000	7.2	0.0146	0.105
32500	7.8	-0.0154	-0.12
35000	8.4	-0.0154	-0.13
37500	9.0	0.009	0.081
40000	9.6	0.009	0.086
42500	10.2	-0.0086	-0.088
45000	10.8	-0.0086	-0.093
47500	11.4	0.0094	0.107

To a large extent, the Periodic Burst System could be implemented with existing units. The only area of development uncertainty is in regard to synchronization of the Burst Decommutator.

It is recommended that the Burst Study be continued in two areas:

(1) Develop and fabricate an engineering model of the Burst Decommutator Synchronizer (discussed in section 2.2.2.3) and establish the performance boundaries of that equipment.

(2) Accomplish a design study which would result in detailed specifications of a Periodic Burst System and specify by nomenclature, manufacturer and model number all equipment which could be used as part of the system.

3. Comparison of Periodic and Blocked Burst Techniques

The Periodic Burst Technique has been described in the preceding section of this report, and the Blocked Technique is described in Appendix A. In this section, characteristics of the two techniques, which are considered significant from an operational or implementation viewpoint, are compared.

3.1 Time Delay

The time which elapses between the selection of a sample and the time it is displayed is a delay. This delay results from propagation time, machine operating times, but mostly to the amount of time the sample is held in memories. In the Periodic system the maximum length of time a sample is held in storage before transmission is one period at the data rate. At the display terminal of the system the storage time is very small and the total delay in a system of this type is in the order of 1 period at the particular data rate. In the Blocked Technique, a parameter may remain in storage as long as 0.25 seconds before it is transmitted and may again be stored for up to 0.25 seconds before being read-out for display. Hence, a total delay of up to 0.5 seconds is possible.

3.2 Memory Sizes

The total memory required for the Periodic Technique was computed in 2.7.3.7, to be 151,315 bits. In the Blocked system, it was found that 1084 words (8672 bits) was needed for the storage of parameters at the transmission terminal. At least that many more would be needed for addresses. At the decommutation terminal, at least 8672 bits would be needed to store decommutated parameters until they can be read-out for display. In addition, a Look-Up Table memory, which would correlate the identity and sampling times of parameters in a block with the input links from which they were selected, would be required. This Look-Up Table would be approximately the same size as the one used in the Periodic Technique. It is estimated that the total memory requirement of the Blocked system would be about 10,000 bits larger than that of the Periodic system, or 161,000 bits.

3.3 Bandwidth Utilization Efficiency

The Periodic Burst system was found to achieve a BUE of 87.8% for the particular conditions (Titan, Gemini and Agena) vs 96.0% for the Blocked System. It is estimated that the maximum practical BUE for the periodic Burst system would be about 95%. The Blocked System is thought to have achieved about its maximum possible (96%) in the specific case examined. Other conditions could cause that achievement to be reduced, and it is expected that BUE's of about 90% may be realistic for an arbitrary combination of input links and communications circuits.

3.4 Recommendations

The Periodic Burst Technique is recommended.

APPENDIX A

BLOCKED BURST TECHNIQUE

Blocked "System" Summary

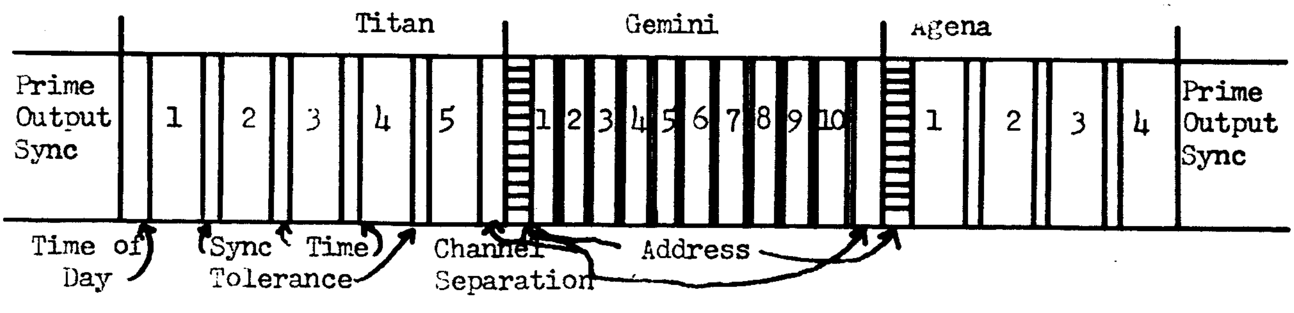
The Block Burst retransmission system receives data from 3 input links- Titan, Gemini and Agena, and provides an output in block form at a bit and frame rate which is different from the input links. The rates are listed as follows:

	Bit Rate	Frame Rate
Titan Input	172.8 kb/s	20 f/s
Gemini Input	51.2 kb/s	40 f/s
Agena Input	16.384 kb/s	16 f/s
Burst Output	40.8 kb/s	4 f/s

The quantity of available input data far exceeds the retransmission capability, therefore, a selection of input data to be retransmitted must be made. The main considerations of a retransmission system which must provide time of occurrence along with the data parameters, are data time delay and bandwidth utilization efficienty (B.U.E.). The data time delay is the time between data occurrence at the input on the retransmission system and data availability to the user. B.U.E. is a measure of maximum possible output data based on the output data rate versus the actual output data. Each output frame contains three blocks, one from each of the three asynchronous Titan, Gemini and Agena telemetry links. A block will contain all the data samples selected from a given input link over some period of time. That period will contain an integer number of input frames. In this particular system it has been found that a minimum output period equals 0.25 seconds, and contains three blocks, one from each input link. Each block is also assembled during an 0.25 second interval and contains data selected from 5 Titan input frames, 10 Gemini input frames, and 4 Agena input frames.

Memory is required to store selected data samples until the block is readout. There is some overlap where read-in and read-out simultaneously occur, therefore, a full 0.25 seconds of samples do not have to be stored. For an assumed data mix of 147 Titan word/frame, 50 Gemini word/frame and 10 Agena word/frame, a memory of 1080 word capacity is required.

A typical arrangement of the format words in the output frame is shown in the following figure:



The time of day word will contain units of hours, minutes, seconds, and micro seconds. The time word indicates the time of occurrence at the receiver terminal of the retransmission system of the first input data frame in output frame. In the case shown, it would be the time of the first Titan frame in the Titan block. Once time is established at the start of the output frame, all data within the output frame will be referenced to this time. The time of the first frame of each block is determined from a channel separation word. That word is a measure of the delay between the transmitted time of day word and the beginning of the first frame of the other two blocks.

The time of occurrence of parameters selected during subsequent input frames and assembled into a particular block, can be reestablished from a "Sync Tolerance Word" which is a measure of the difference between the actual input sync rate and the nominal input sync rate. The address words are used to identify the prime input frame number. Since the input frames following the first frame of each block are in numerical ascending order, it is only necessary to identify the first frame of each block.

A.1 Establishing the Burst Period

A Burst period is defined as the time in which all inputs have cycled through the smallest number of complete frames. Thus:

$$\frac{N_1}{f_1} = \frac{N_2}{f_2} = \frac{N_3}{f_3} = \frac{N_4}{f_4} = \frac{N_n}{f_n} = P$$

where N = number of complete frames

f = main frame rate;for three input only

$$N_1 = \frac{f_1}{f_2} N_2 = \frac{f_1}{f_3} N_3$$

$$N_2 = \frac{f_2}{f_3} N_3$$

for the case of Titan, Gemini, and Agena.

where Titan = f_1 = 20 frames/sec.
Gemini = f_2 = 40 "
Agena = f_3 = 16 "

$$N_1 = \frac{20}{40} N_2 = \frac{N_2}{2}$$

$$N_1 = \frac{20}{16} N_3 = \frac{5}{4} N_3$$

$$N_2 = \frac{40}{16} N_3 = \frac{5}{2} N_3$$

An output period will contain three blocks, each selected over an integer number of input frames. Hence, N_1, N_2, and N_3 must be integers. By inspection of the above equations, it is seen that a satisfactory solution is obtained when:

$$N_1 = 5$$

$$N_2 = 10$$

$$N_3 = 4$$

and:

$$P = \frac{5}{20} = \frac{10}{40} = \frac{4}{16} = 0.25 \text{ sec.}$$

A.2 Determination of Maximum Number of Selectable Parameters

The number of parameters which can be selected from each input frame of a given input link is equal to the retransmission bandwidth allocated to that input link, divided by the input frame rate and word length.

It is obvious that in an output period of 0.25 seconds, the total number of words which can be transmitted is 40.8x0.25/8 = 1275. If these were all Titan words it would be allowable to select 1275/5 = 225 words per input frame; if all Gemini words, 1275/10 = 127.5 words per input frame; if all Agena, 1275/4 = 318.75 words per input frame.

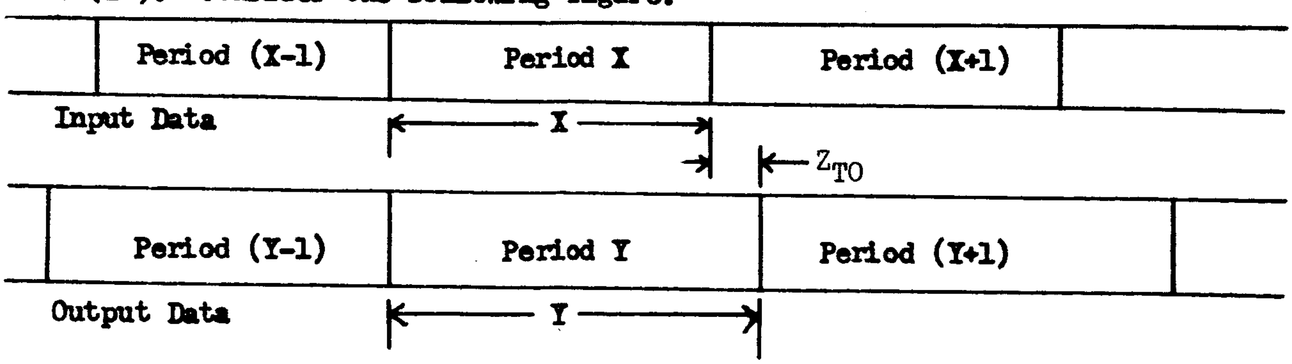
In a more representative condition, where blocks from all input links are included in the output, it is evident that fewer words can be selected from each input frame. It is assumed that the output capacity is equally divided between the three input links, each will use 425 words per block. The number of words which can be selected per input frame is then:

Titan, 425/5 = 85; Gemini 425/10 = 42.5; Agena 425/4 = 106.25.

It is evident that other mixes are possible.

A.3 Effect of Input and Output Link Tolerances

Consider the case where only the Titan link is used. If the data input period is less than the data input period, then the difference in time multiplied by the input selected data rate (not to exceed 255 words/frame) would be the overflow (Zo). Consider the following figure:



The data overflow (Z_o) is determined as follows:

$$Z_{TO} = \text{time difference} = Y - X$$

Overflow (Z_o) = $R_n \times Z_{TO}$ where R_n is the input data rate and Z_o is not to exceed 255 words/frame.

If the data input period is greater than the data output period, then the difference in time multiplied by the output data would be the underflow. Consider the following figure:

Input Data

	Period (X-1)	Period X	Period (X+1)	

Output Data → ← Z_{TU}

	Period (Y-1)	Period Y	Period (Y+1)	

The data underflow (Z_U) is determined as follows:

$$Z_{TU} = \text{time difference} = X - Y$$

$$\text{Data Underflow } (Z_U) = R_O \times Z_{TU}$$

where R_o is the output data rate.

Lastly, the third and most ideal case is when the data underflow or data overflow equals zero, however, tolerances of the input and output links makes this condition for all practicable purposes impossible. Since underflow of data is allowable (no loss of data occurs), the maximum number of words selected must be based on the maximum input data rate and the minimum output data rate.

Consider that the input link has a rate tolerance of ± .05% of 172 Kb/s, in terms of bits/sec., this is ± 90 bits/sec. Also, considering that the output link has a rate tolerance of ± .01%, since it is a ground link, or ± 4.0 bits/sec. for all practicable purposes the tolerance of the output link can be neglected as insignificant.

In a condition where the incoming data rate is a minimum of (172.8-.09)Kb/s. The maximum utilization loss of output data locations is the difference between the maximum input data rate and the minimum input data rate multiplied by the ratio of the output data rate divided by the maximum input data rate, or expression of the utilization on loss due to rate tolerances in equation form.

$$U.L._{(R)} = (\text{maximum input rate} - \text{minimum input rate}) \times \frac{\text{Output rate}}{\text{Max. In.}}$$

$$= (172.89 \text{ Kb/s} - 172.71 \text{ Kb/s}) \frac{40.8 \text{ Kb/s}}{172.89 \text{ Kb/s}}$$

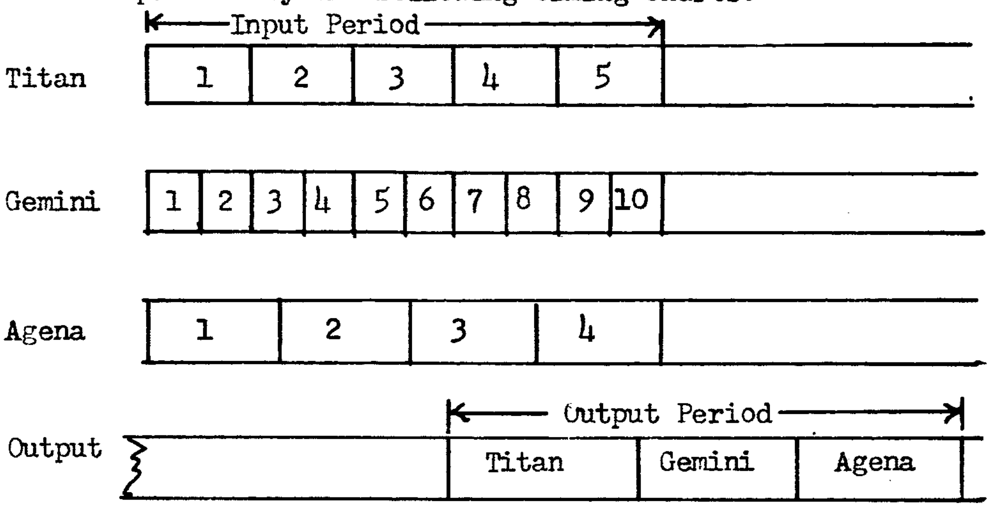
$$= 180 \times \frac{40.8}{172.89} \cong 44 \text{ bits/sec. maximum}$$

or 11 bits/output frame.

The above discussion has only considered the Titan link, however, it can be readily seen that the combinations of Titan, Gemini, and Agena utilization loss will be less than the utilization loss of Titan alone.

A.4 MEMORY REQUIREMENTS

The question might be asked as to whether we need storage for Titan input data during Gemini and Agena transmission this can best be explained by the following timing charts:



If it were required for readout to occur at the leading edge of frame 1 of each channel then it would be necessary for additional storage. However, since the selected input period is a common multiple of all channels it is possible to move the frame slots in phase and still be in time-lock. This would mean that the start of the period would occur at some time other than the leading edge of a frame or the first frame after input channel sync. This can be best shown on the following chart.

Where ϕ, B & α is the phase shift sync of each channel burst and the phase lock time and ϕ'. B' & α' is the phase shift between the sync prior to phase lock of each channel and the phase lock time. The period of complete cycles of all channels is τ.

It can be seen that though the storage is dependent only on the input word rate and the number of input words per frame, extra storage for buffering between other channels rates is not necessary as long as each channel has a block size which is independent of output frame rate for any one program. Constant but non-related block sizes allow the input channels to be phased locked thus all channels will track together and overflow or underflow of any one channel will be a function of variation in the input rate of that channel and constant overflow or underflow of all channels equally will be a function of the variation in the output rate only.

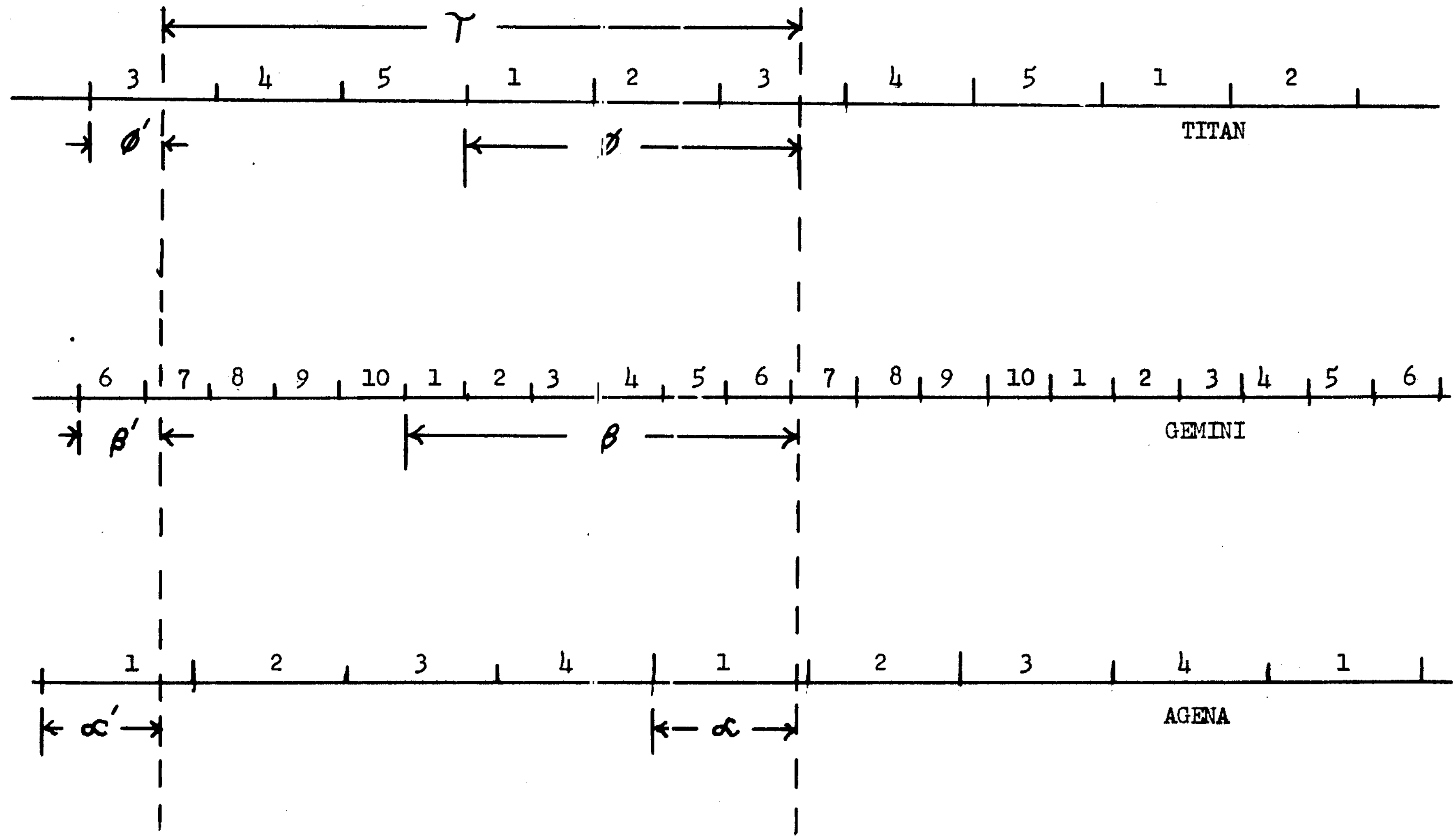
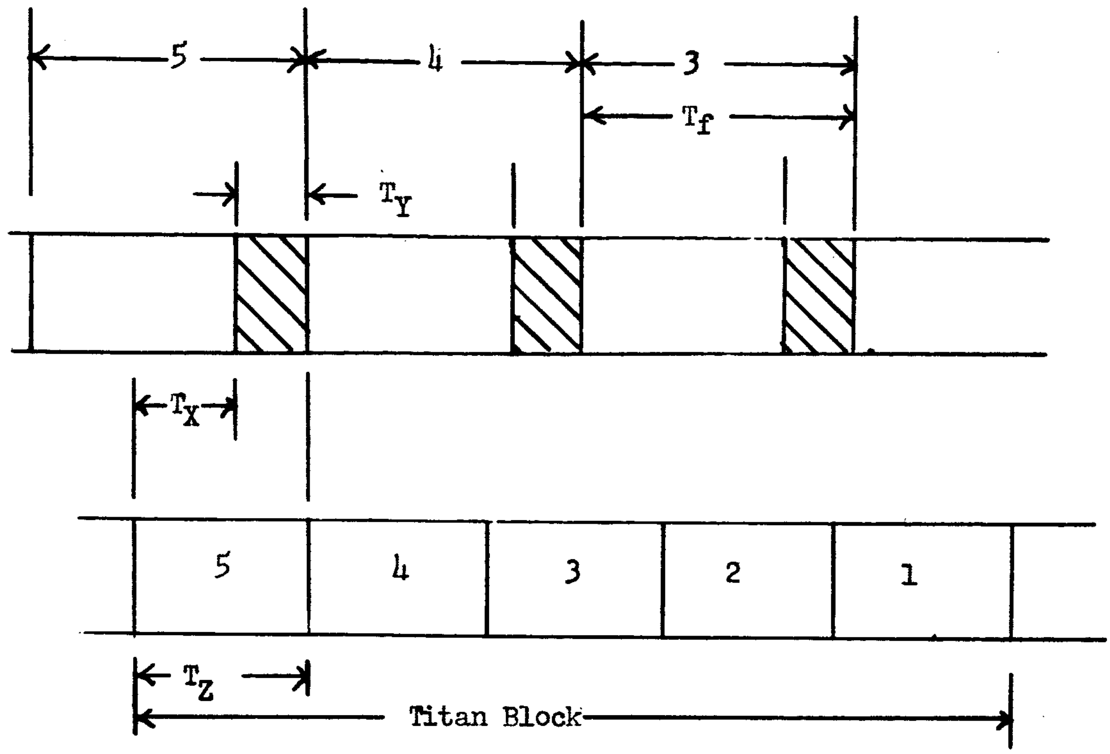


FIG. A-1 PHASE RELATIONSHIP OF VARIOUS INPUT LINKS

A.4.1 DETERMINE TIME BETWEEN END OF LOAD AND END OF UNLOAD(T_x)



The time required to unload an input frame is labeled T_Z, this time less the time to load (T_Y) is the time (T_x) between the end of load and the end of unload. This time must be included when calculating total buffer storage. Solving for T_x

$$T_X = T_Z - T_Y \qquad \text{where:} \quad T_X \quad T_f$$

The maximum storage requirement occurs when the selected data words occur in a burst. Since this condition may occur independant of the equipment and as a function under operational control. The maximum storage condition can be calculated.

Terms of selected words/frame and input and output data rates as follows:

$$T_z = \frac{1}{R_o} \times {}^{\text{bits}}/_{\text{word}} \times {}^{\text{words}}/_{\text{frame}}$$

$$T_y = \frac{1}{R_n} \times {}^{\text{bits}}/_{\text{word}} \times {}^{\text{words}}/_{\text{frame}}$$

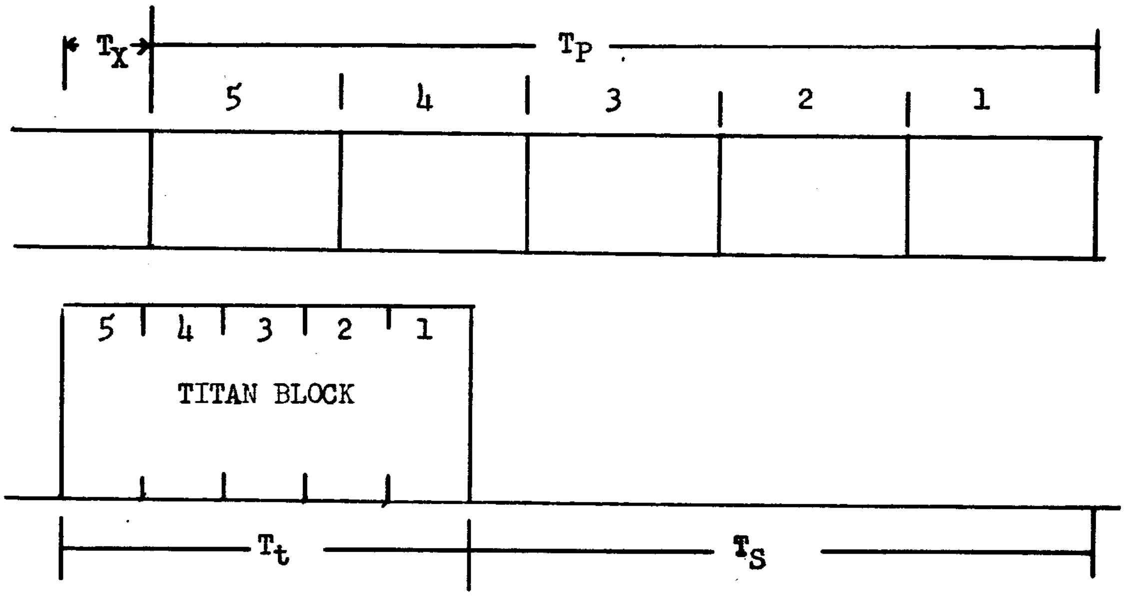
$$T_x = T_z - T_y = {}^{\text{words}}/_{\text{frame}} \times {}^{\text{bits}}/_{\text{word}} \times \left(\frac{1}{R_o} - \frac{1}{R_n}\right)$$

$$T_x = (8) \quad {}^{\text{words}}/_{\text{frame}} \left(\frac{1}{R_o} - \frac{1}{R_n}\right)$$

where R_o = output data into in $^{bits}/_{sec}$

R_n = input data into in $^{bits}/_{sec}$

A.4.2 DETERMINE DELAY REQUIRED TO ASSURE ALL SAMPLES OCCURRING



T_S is the time required before transmission can start so that input data words are received before their re-transmission time, and can be computed as follows:

The worse case occurs when the selected input data words occurs in a burst at the leading edge of the data frame, here again this conditions is a function of operational control and therefore must be taken into account in the determination of buffer storage requirements.

Referring to the timing chart, the time before transmission is determined as follows:

$$T_S = T_P + T_X - T_t$$

Since the Titan data is asynchronous , it is impossible to control the frame with respect to data output, therefore worse case analysis clearly indicates that the Titan data frame could slip one complete frame; as a result the time before transmission is

$$T_S = T_P + T_X + T_{(slip)} - T_t$$

$$T_P = \frac{1}{F_R} \times F = \frac{1}{4} \text{ sec.}$$

$$T_t = \frac{1}{R_o} \ \frac{\text{words}}{\text{frame}} \ (\text{Frames}) \ \frac{\text{bits}}{\text{word}} = (10^{-4}) \ \frac{\text{words}}{\text{frame}} \ \text{sec.}$$

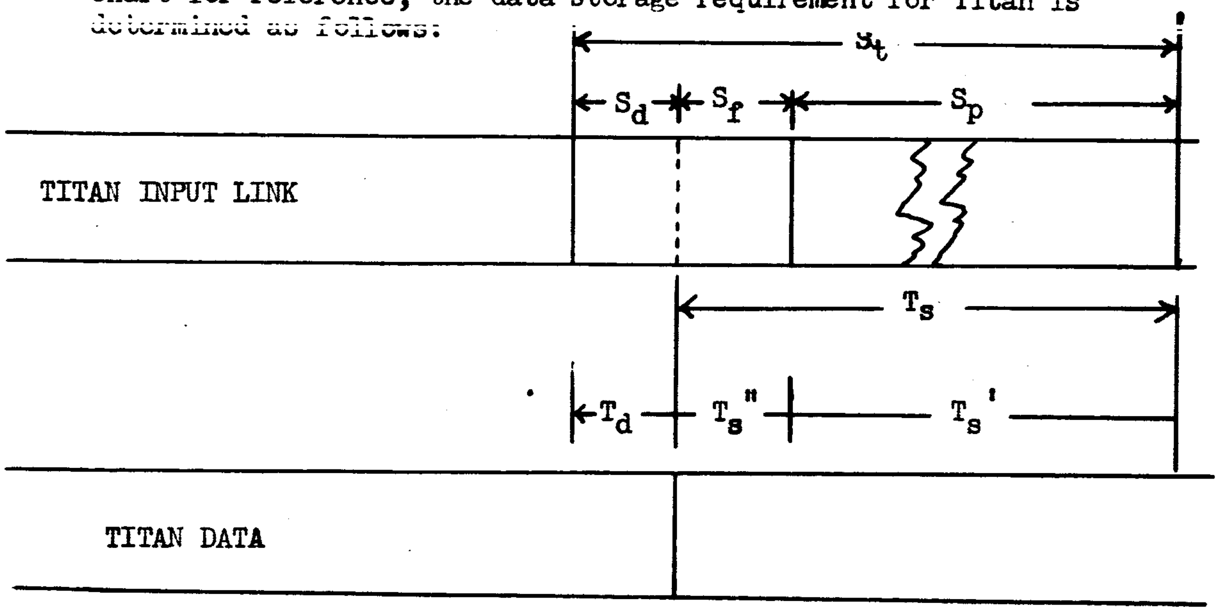
$$T_{(slip)} = \frac{1}{F_R} \times (1 \text{ Frame}) = \frac{1}{20} \text{ sec.}$$

$$T_X = 8 \ \frac{\text{words}}{\text{frame}} \ \frac{1}{R_o} - \frac{1}{R_m} \ \text{sec.}$$

$$T_s = \frac{1}{4} + 8 \ \frac{\text{words}}{\text{frame}} \ \frac{1}{R_o} - \frac{1}{R_m} + \frac{1}{20} - 40 \ \frac{\text{words}}{\text{frame}} \ \frac{1}{R_o}$$

A.4.3 STORAGE REQUIRED FOR A SINGLE INPUT LINK

The data storage requirement in bits is dependent on how much Titan data is sent in time T_s. Referring to the following timing chart for reference, the data storage requirement for Titan is determined as follows:



where 1) S_p is the storage required for data received from Titan in complete frames

2) S_f is the storage required for data received from Titan in fractions of a complete frame but not to exceed the total words/frame storage requirements.

3) S_d is the storage required for data received from titan while data is being transmitted but not to exceed (words/frame - S_f).

$$S_p = T_s{}' \times \frac{\text{sync rate}}{\text{period}} \quad \text{(Frames)}$$

$$S_p = (T_s{}'')\ (R_{in})\ \frac{\text{words}}{\text{bit}} \quad \text{and } S_f \quad \frac{\text{words}}{\text{frame}}$$

$$S_d = \frac{\text{words}}{\text{frame}} - S_f \quad \frac{R_{in} - R_o}{R_{in}}$$

where: $T_s{}'\ S_p$ is whole part of $T_s S_p$

and: $T_s{}'' = (T_s - T_s{}')$

$$S_T = (T_s{}')\ \frac{\text{sync rate}}{\text{period}}\ \frac{\text{words}}{\text{frame}} + (T_s{}'')\ (R_n)\ \frac{\text{words}}{\text{bit}} + \frac{\text{words}}{\text{frame}} - S_f \quad \frac{R_n - R_o}{R_n}$$

Similarly Gemini Storage is:

$$S_G = (T_s{}')\ \frac{\text{sync rate}}{\text{period}}\ \frac{\text{words}}{\text{frame}} + (T_s{}''\ R_n)\ \frac{\text{words}}{\text{bit}} + \frac{\text{words}}{\text{frame}} - S_f) \quad \frac{R_n - R_o}{R_n}$$

The total storage for Agena is somewhat different than with Titan or Gemini. Since the Agena output data rate is less than the Agena input data rate, the values for T_x and S_d are zero:

$$S_A = T_s{}' - (\text{sync rate}) \frac{\text{words}}{\text{frame}} + (T_s{}'' R_n) \frac{\text{words}}{\text{bit}}$$

A.4.4 STORAGE REQUIRED FOR THREE SIMULTANEOUS LINKS

The total memory storage required for all channels is:

$$S = S_T + S_G + S_A$$

$$\text{Total storage} = T_s{}' S_R W_f + \frac{T_s{}'' R_n}{8} + (W_f - S_f) \frac{R_n - R_o}{R_n}$$

$$+ T_s{}' S_R W_f + \frac{T_s{}'' R_n}{8} + (W_f - S_f) \frac{R_n - R_o}{R_n}$$

$$+ T_s{}' S_R W_f + \frac{T_s{}'' R_n}{8}$$

where 1) values of parameter are based on individual channel constants.

2) S_R = Sync Rate for the particular channel under discussion

3) W_f = words/frame for the particular channel under discussion

4) f = frames/period for the particular channel under discussion.

$$W_{f(\text{totals})} = (fW_f)_T = (fW_f)_G + (fW_f)_A$$

A.4.5 DETERMINING STORAGE FOR PARTICULAR OUTPUT MIXES

It is obvious that maximum storage does not occur when the total words are allocated for use by the Agena channel, because S_d and T_x are zero, and in addition the input data rate R_n is very low with respect to the Titan input data rate.

Considering now Titan word assignment against Gemini word assignments. The input data rate R_n of Titan is much greater than Gemini, and in addition the sync period for Titan is much greater than Gemini.

A.4.5.1 ALL TITAN DATA

The total memory storage required when the total words/period W_f(total) is assigned to handle Titan data only is determined as follows:

$$\text{Total Storage} = T_s{}' \, S_R \, W_{f(T)} + \frac{T_s{}'' \, R_n}{8}$$

$$+ (W_{f(T)} - S_f) \, \frac{R_n - R_o}{R_n}$$

The total word capacity of the output submarine cable (W_f(total)) for one period is dependent on the output data rate.

$$W_f \text{ (Total)} = \text{(bit rate)} \left(\frac{1}{\frac{\text{period}}{\text{sec}}} \right) \left(\frac{\text{word}}{\text{bits}} \right)$$

$$= \frac{R_o P}{8}$$

$$= \frac{40800}{4 \text{x} 8} = 1{,}275 \text{ words}$$

Given : $S_R = \dfrac{1}{20}$

Given $R_n = 172$ KBPS

Given $R_o = 40.8$ KBPS

$f = 5$

$$T_x = 8 \text{ x } 255 \left(\frac{1}{40800} - \frac{1}{172{,}000} \right)$$

$$= 8 \text{ x } 255 \ (0.0000251 - 0.0000058)$$

$$= 0.0394 \text{ sec.}$$

$$T_T = \frac{40\text{x}255}{R_o} = \frac{40\text{x}255}{40800} = .256 \text{ sec.}$$

$$T_s = T_p + T_x\, T(\text{slip}) - T_T$$

$$T_s = 0.25 - 0.0394 + 0.05 - 0.256 = 0.08$$

$$T_s S_R = 0.08 \times 20 = 1.6$$

$$T_s{'} S_R = 1$$

$$T_s{''} = \frac{0.6}{S_R} = \frac{0.6}{20} = 0.03 \text{ sec}$$

$$\text{Total Storage} = 1.0 \times 255 + \frac{.03 \times 172K}{8}$$

The second term is

$$S_f = \frac{.03 \times 172K}{8}$$ which is greater than the words/frame therefore

$S_f = 255$ which is the maximum words/frame

$$\text{Total Storage} = 1 \times 255 + 255$$
$$= 510 \text{ words}$$

This is not the maximum storage since variations in T'_s was not considered.

A.4.5.2 SELECTED MIXTURE OF AGENA, GEMINI, AND TITAN DATA

Consider a second case using the following parameters:

$$W_{f(t)} = 147 \times 5 = 735$$

$$W_{f(G)} = 50 \times 10 = 500$$

$$W_{f(A)} = 10 \times 4 = 40$$

$$\text{TITAN STORAGE} = T'_s\, S_R W_{f(T)} + \frac{T''R_n}{8} + \left(W_{f(T)} - \frac{T''R_n}{8}\right)\left(\frac{R_n - R_o}{R_n}\right)$$

Where: $\frac{T''R_n}{8}$ must equal an integer

and: $W_{f(T)} - \frac{T''R_n}{8} \geq 0$

$$T_x = 8 \times 147 \times \left(\frac{1}{40800} - \frac{1}{172000}\right) = 8 \times 147 \times 0.0193 \times 10^{-3} = 0.0227 \text{ sec}$$

$$T_T = \frac{40 \times 147}{40800} = 0.1476 \text{ sec}$$

$$T_s = 0.25 + 0.0227 + 0.05 - 0.1476 = 0.1754 \text{ sec.}$$

$$T_s R_s = 0.1754 \times 20 = 3.508$$

$$T_s' R = 3.0$$

$$T'' = \frac{0.508}{20} \approx 0.02$$

$$S_f = \frac{0.02 \times 172000}{8} \approx 430 \text{ which is} > 147$$

$\therefore$ TITAN STORAGE = $(3 \times 147) + 147 = 588$ words

$$\text{GEMINI STORAGE} = T'_s S_R W_{f(G)} + \frac{T''R_n}{8} + \left(W_{f(G)} - \frac{T''R_n}{8}\right) \frac{R_n - R_o}{R_n}$$

$$T_x = 8 \times 50 \left[(0.0251 - 0.0195)10^{-3}\right] = 0.00224$$

$$T_T = \frac{50 \times 5 \times 8}{R_o} = 0.0502$$

$$T_s = 0.25 + 0.0022 + 0.025 - 0.0502 = 0.227$$

$$T_sR_s = 0.227 \times 40 = 9.08$$

$$T_s' R = 9$$

$$T'' = \frac{0.08}{40} = 0.002$$

$$S_f = \frac{0.002 \times 51,200}{8} = 13 \text{ words}$$

$$\text{GEMINI STORAGE} = (9 \times 50) + 13 + (50 - 13)\left(\frac{51.2 - 40.8}{51.2}\right)$$

$$= 400 + 13 + 37\ (0.0195)(10^{-4})$$

$$= 421 \text{ words}$$

$$\text{AGENA STORAGE} = T'S_R\ W_{f(A)} + \frac{T''_s\ R_n}{8}$$

$$T_x = 0$$

$$T_T = \frac{10 \times 16 \times 8}{R_o} = \frac{1280}{16380} = 0.0781$$

$$T_s = 0.25 + 0.0625 - 0.0781 = 0.2344$$

$$T_sR_s = 0.2344 \times 20 \approx 4.7$$

$$T'R = 7$$

$$T'' = \frac{0.7}{16} -$$

$$S_f = \frac{0.7}{16} \times \frac{16.38}{8} = 1$$

$$\text{AGENA STORAGE} = (7 \times 10) + 1 = 71$$

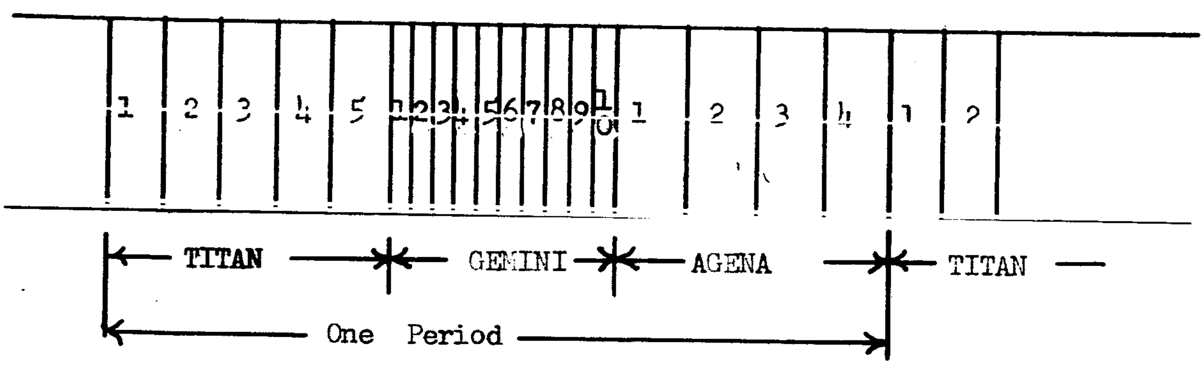
$$\text{TOTAL STORAGE} = T + G + A = 588 + 421 + 71 = 1080 \text{ words}$$

A.5 CRITERIA FOR CORRELATING ASYNCHRONOUS DATA

The criteria used to develop the Burst message format is based on the ability to establish a relationship between the Titan, Gemini and Agena prime frame. The minimum period for synchronous data transfer as covered in a previous section of the report is 1/4 second. It is important to note that the data is not locked in sync and though it is phased locked, memory locations are provided for frame slippage as discussed earlier. A complete period of data contains 5 blocks of Titan data, 10 blocks of Gemini data and 4 blocks of Agena data, each block represents a prime frame of the incoming data. So far in the discussion we are assuming that the format from one period to the next is identical in every respect except some slot locations of the blocks are multiplex to send more than one parameter per slot. This case is when parameters have lower sampling rates than the prime block rate. The above configuration requires time and address tags for every parameter. However, since the parameter time and location is known with relationship to each block, only block time and subframe address will be required.

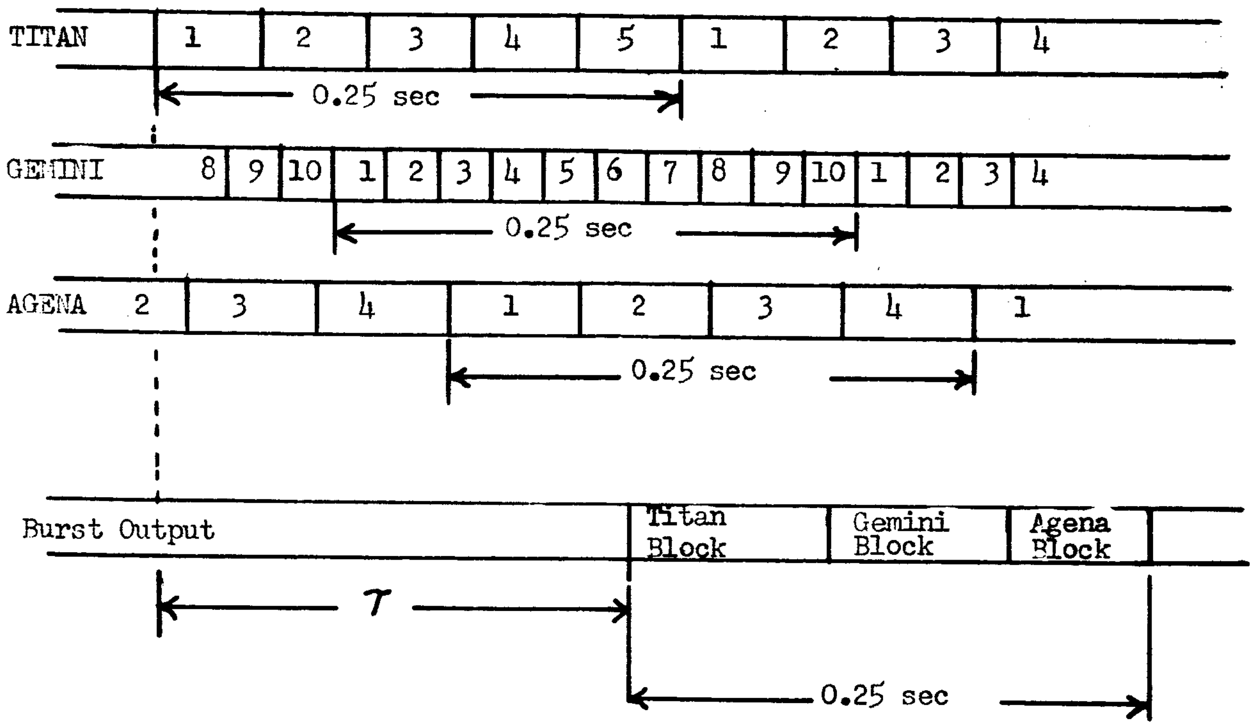
A.5.1 TECHNIQUES FOR TIME CORRELATING OF ASYNCHRONOUS LINKS

The block format of the period is arranged as shown below



In order to simplify the programming, the block sizes of any one channel will be constant for any one program, however, the block sizes of different channels may vary depending on the number of parameters required from the channel. A comparison will be made later in this report between the saving if any, in output data location versus the resulting program complexity as a result of varying block sizes within a channel.

The first five blocks contains Titan parameters followed by ten blocks of Gemini parameters and in turn followed by four blocks of Agena parameters. The order of location of Titan, Gemini and Agena channels within the period as shown in the block format is not critical as long as all the blocks of any one channel are located together. For ease of programming, however, the first block of a period should be selected such that its prime sync will occur before the prime sync of any of the following blocks in the period. This allows every block in the period to be referenced to the time of occurance of the first block. This can best be shown by the following timing diagram.



A.5.1.1 ALLOCATION OF TIME FORMAT BITS

The time of occurrence of the first sync within the period is detected and stored in memory. The time word will contain units of hours, minutes and seconds and will be measured to an accuracy of one microsecond. The number of bit location required for the complete time word is tabulated as follows:

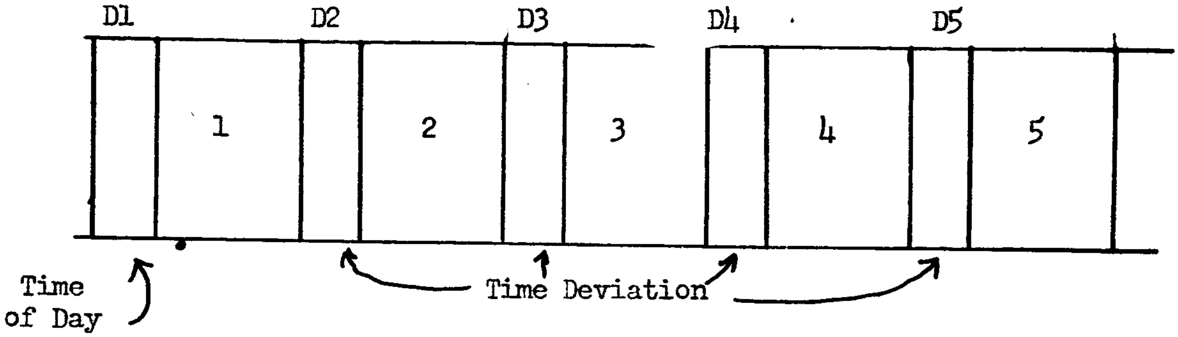
Number System	Bit Location Hours	Min.	Sec.	Total Bits
1. Binary Coded decimal	6	7	31	44
2. Straight binary with separation between units of hours, min., sec.	5	6	26	37
3. Straight binary with no separation				37

The number of bits listed above are based on a maximum number of 24 hours, 60 minutes and 60 seconds.

Since the primary interest is in transmitting the maximum data within a period rather than standardizing the time format. The study will consider only the 2nd system which uses the least number of output data bit locations, yet has some form of time separation.

A.5.1.2 CONSIDERATION OF DATA RATE TOLERANCE FOR SINGLE LINKS

Once time is established at the start of a block it is only necessary to determine the deviation in the known input data rate in order to determine "time of occurrence" at any point. The proposed format is shown in the following Titan output block diagram.



The time deviation of sync for a given frame from normal is formulated and placed at the beginning of the given data block. The Maximum Time Data Block #2 can be displaced is ±25 usec from Block #1, as indicated below, therefore only 6 time deviation bits are needed, one for sign and 5 for quantity. Block #3 can be displaced up to ±50 usec from block one's time of day and so on for blocks #4 and #5. The bit location for time deviation are located at D2, D3, D4, and D5.

The data rate tolerance of the Titan link is assumed to be .05% when expressed in terms of time is .05% of the .05 second data sync input period. The data rate tolerance for the sync expressed in microseconds is:

Titan sync tolerance in seconds = .0005 x .05 = 25 microseconds/ input period.

Over a full period of output data the tolerance is 125 microseconds determined by multiplying the sync tolerance by the number of sync periods per output periods. If we assume this tolerance is a fair representation of all channels, then the maximum deviation between the start of a period and the end of a period is 125 microseconds. The tolerance in microseconds for each frame is listed below:

Frame #1	Time of day
#2	Sync tolerance x 1 (+25 usec)
#3	Sync tolerance x 2 (±50 usec)
#4	Sync tolerance x 3 (±75 usec)
#5	Sync tolerance x 4 (±100 usec)

The bits required for each location is as follows:

Titan Location D1 (Time of Sync)	-	37 bits
Titan Location D2 (±25 usec deviation)	-	6 bits
Titan Location D3 (±50 usec deviation)	-	7 bits
Titan Location D4 (±75 usec deviation)	-	8 bits
Titan Location D4 (±100 usec deviation)	-	8 bits
		66

A.5.1.3 MULTIPLE-LINK SYNC TIME TOLERANCE CONSIDERATION

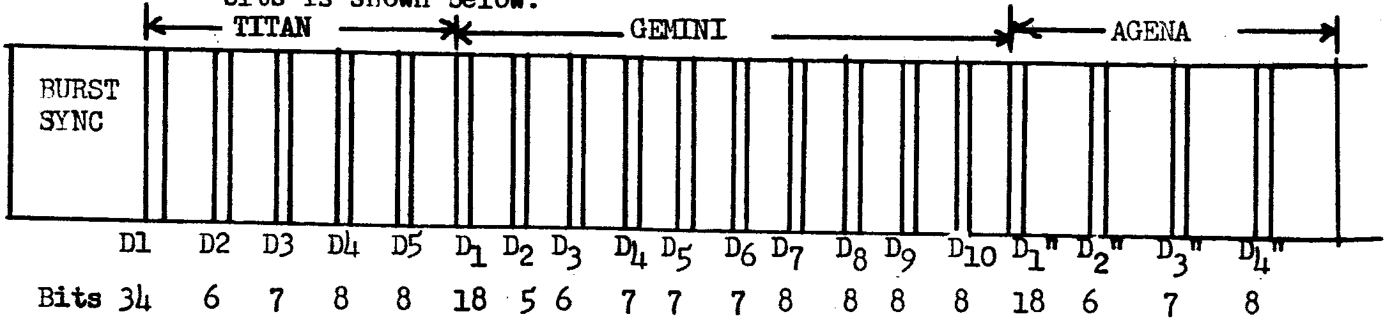
Gemini and Agena Channels could be handled similar to Titan using a 34 bit time of sync word followed by frame or block time of sync occurs after the first Titan sync it is possible to reference Gemini and Agena time of sync to the Titan time of sync, sending only the time of first sync difference of Gemini and Agena instead of the actual time. The maximum difference is .25 seconds or 250,000 microseconds. Thus requiring only 18 bits instead of the 34 need for sending time of sync.

All Gemini and Agena blocks other than the first are formatted in the manner similar to the Titan blocks. The deviation times for Gemini and Agena for .05% tolerance are listed as follows:

Gemini Location D^1_1 (Time of Sync Difference) +250,000 usec max.	-	18 bits
Gemini Location D^1_2 (±13 usec deviation)	-	5 bits
Gemini Location D^1_3 (±25 usec deviation)	-	6 bits
Gemini Location D^1_4 (±38 usec deviation)	-	7 bits
Gemini Location D^1_5 (±50 usec deviation)	-	7 bits
Gemini Location D^1_6 (±63 usec deviation)	-	7 bits
Gemini Location D^1_7 (±75 usec deviation)	-	8 bits
Gemini Location D^1_8 (±88 usec deviation)	-	8 bits
Gemini Location D^1_9 (±100 usec deviation)	-	8 bits
Gemini Location D^1_{10} (±113 usec deviation)	-	8 bits
		82

Agena Location D''_1 (Time of Sync Difference) +250,000 usec max.	-	18 bits
Agena Location D''_2 (±32 usec deviation)	-	6 bits
Agena Location D''_3 (±63 usec deviation)	-	7 bits
Agena Location D''_4 (±94 usec deviation)	-	8 bits
		39

The complete output period diagram showing the location of timing bits is shown below:



The total bits needed to time correlate all the data words with a data input tolerance of .05%

Titan	-	66
Gemini	-	82
Agena	-	39

Total Time correlation bits 187 bits

converting to 8 bit words

Total time correlation words = 24 words plus 5 spare/bits

A.5.1.4 VARIATION OF SYNC TIME BITS VERSUS CHANGE IN SYNC. FREQUENCY TOLERANCE

If the Tolerance is reduced then the sync time deviation bits could be reduced, however the amount of reduction would be small in comparison to the reduction in tolerance. As an example consider an input channel tolerance of .01% instead of .05%, the maximum sync deviation per output frame would be determined as follows:

Maximum Titan Sync Deviation = (.01% x input frame period)

= .0001 x .05 = 5 microseconds

The following bits would be required for time correlation of all data:

Titan

D_1 (Time of Sync)	37	D1'(Sync difference)	18	D1"(Sync diff)	18
D2 ± usec deviation	4	D2' ±3 usec deviation	3	D2"±7 usec deviation	4
D3 ± usec deviation	5	D3' ±5usec deviation	4	D3"±13 usec "	5
D4 ±15 usec deviation	5	D4' ±8 usec deviation	5	D4"±20 usec "	6
D5 ±20 usec deviation	6	D5' ±10usec deviation	5		
		D6' ±13 usec deviation	5		
		D7' ±15 usec deviation	5		

D8' ± 18 usec deviation 6

D9' ± 20 usec deviation 6

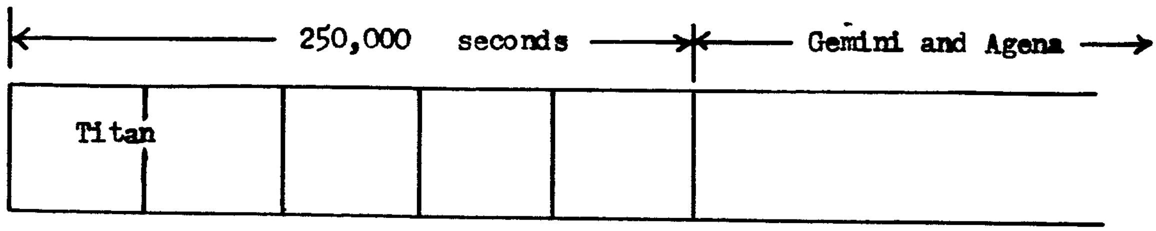
D10' ± 23 usec deviation 6

57	63	33

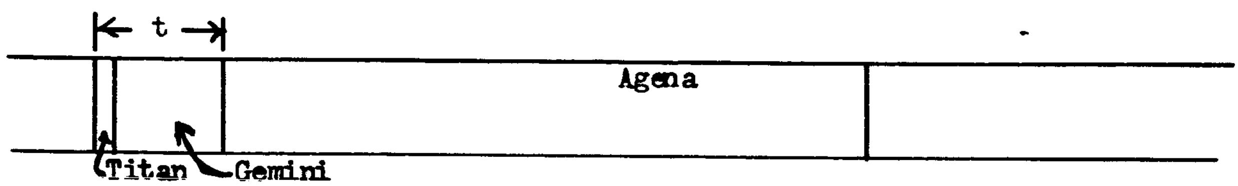
Total time correlation words = $\frac{57 + 63 + 33}{8} = \frac{153}{8}$ = 17 words plus one bit or 20 words with 7 spares. Therefore when accuracy of the input sync is increased from .05% to .01% the words needed for time correlation decrease from 24 to 20. It is obvious that the time correlation words for the proposed system is relatively independent of input channel sync tolerance.

A.5.2 Variation of Time Separations Between First Sync, of Different Channels

This is relatively independent of input channel sync tolerance. The maximum time separation of sync difference between the first Titan sync and the first Gemini sync and between the first Titan sync and first Agena sync is approximately 250,000 microseconds as shown in the following diagram:

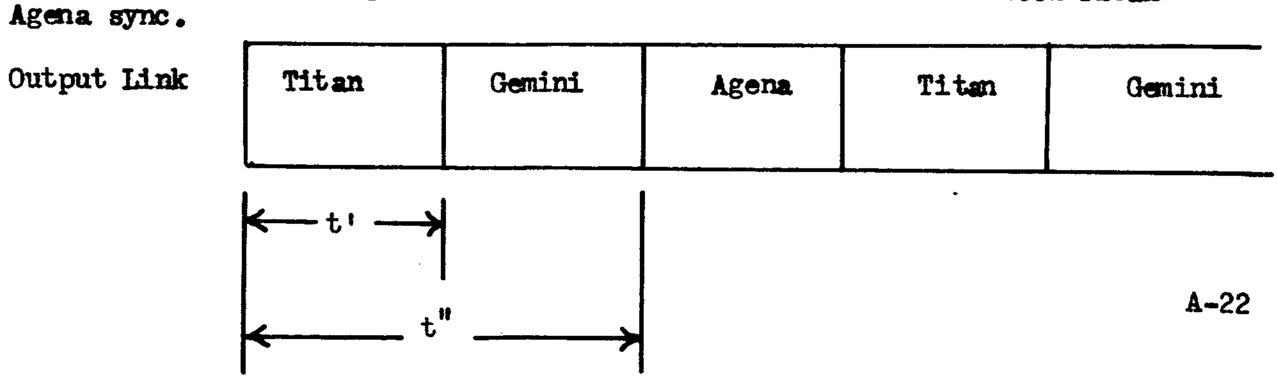


However, this would mean that almost all of the output data is Titan data. Since this case is not a normal case, it is important that we consider other formats of output data. Consider the case where Gemini or Agena data comprised almost all of the output data locations. Then the diagram would be as follows:



The difference between the first Titan sync and the first Gemini or Agena sync(t) would approach zero as a limit. Thus, the word needed for time correlation for an input channel tolerance of .05% would decrease from 24 words ($t \rightarrow 1/4$ sec) to 21 words ($t \rightarrow 0$). Since the two cases mentioned, that is ($t \rightarrow 1/4$ sec) and ($t \rightarrow 0$) are extreme cases, it is worth while to consider the general case where t is a function of the selected words.

In order to simplify the problem and yet still demonstrate the general case, the effect of T_x (page 11), $T_{(slip)}$ (page 14), T_d and T_f (page 15) will be neglected. Thus, the approximate time difference between the first Titan sync and the first Gemini or Agena sync is the time difference measured between Titan Block #1 and Gemini or Agena Block #1 of the output link. This is best shown on the following diagram where t' is the time difference between Titan and Agena sync.



Thus: $t' \approx \frac{\text{sec}}{\text{bits}} \times$ bits/word x Titan word/period

$$t' \approx \frac{1}{40{,}800} \times 8 \text{ bits/word} \times \text{Titan words/period}$$

$$t'' \approx + \frac{\text{sec}}{\text{bits}} \times \text{bits/word} \times \text{Gemini words/period}$$

$$\text{or} \quad t'' \approx \frac{\text{sec}}{\text{bits}} \times \text{bits/word} \times (\text{Titan} + \text{Gemini}) \text{ words/period}$$

The total words needed for timing all output data is approximately 14 words for an input rate tolerance of .05% plus ($t' + t''$) words for sync difference time and words for time of day.

Consider the case where

$$\text{Total } W_f \text{ (Titan)} = 55 \times 5 = 275$$

$$\text{Total } W_f \text{ (Gemini)} = 50 \times 10 = 500$$

$$\text{Total } W_f \text{ (Agena)} = 125 \times 4 = 500$$

$$\text{thus } t' = \frac{8}{40.8K_b} \times 275 = 53{,}900 \text{ usec}$$

$$\text{and } t'' = \frac{8}{40.8K_b} \times (275 \times 500) = \frac{8}{40.8K_b} \times 775 = 152{,}000 \text{ usec}$$

The numbers of bits needed for the above value of t' is 15 and for t'' is 18 or a total of 33 bits for ($t' + t''$) versus a total of 38 for the worse case. It should be obvious to the most casual observer that the above word format weighs heavily in favor of Agena or as in the general case is closer to $t \to 0$ than $t \to .25$ sec.

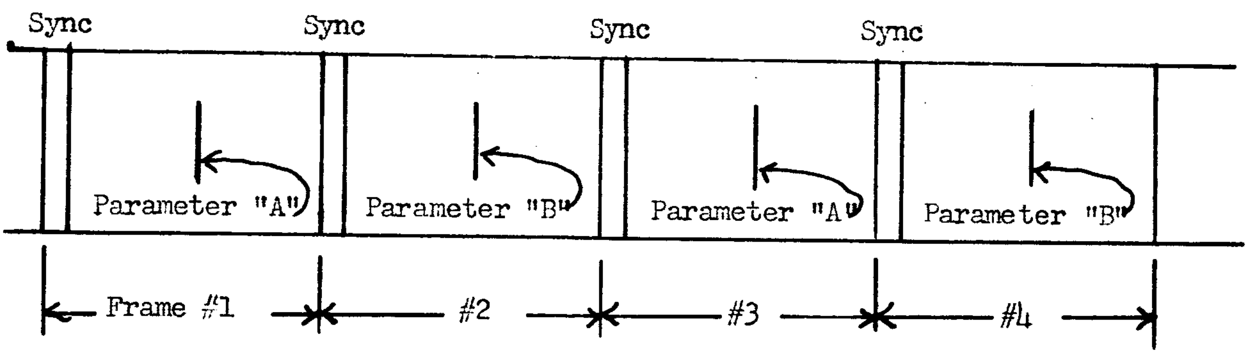
4.5.3 <u>Time and Tolerance Format Conslusions:</u>

It is obvious that only in the extreme cases will any saving occur with respect to sync difference and, therefore, when weighed against program needed to save these extra bit it would indicate that a fixed number of bits be reserved for sync difference and tolerance and this number be such that it covers the worst case. As indicated at the top of page 30, this number should be at least 187 bits or 23 words containing 8 bits/word.

A.5.4 Techniques for Address Correlation of Asynchronous Links

The basic requirement of an asynchronous link system is to be able to determine which data parameter is being sent and its time of occurrence.

In order to provide some form of identification of an input parameter in the output link it is necessary to provide address or identification data in the output data train. If a parameter occurs at the prime frame rate or some multiple of the prime frame rate it is only necessary to identify the sync time of the prime frame. However in cases where the sampling rates are slower it is necessary to determine which prime or main frame contains the data sample. The following diagram gives a pictorial insight to the problem.



If only time of the prime frame occurence were known, it would be impossible to determine if parameter A or B is being sent. In order to locate the slower or sub frame parameters it is necessary to place an identification word or bit in the prime frame containing the slower speed sampling data.

A.5.4.1 Various Known Link Address Correlation Considerations

In the Titan link the slowest sampling rate occurs at the prime frame rate therefore no frame identification is necessary, However, in the Gemini and Agena links, there are sampling rates slower than the prime frame rate, therefore identification bits are necessary. The slowest sub frame of the Geminin link is .416 samples per second, therefore identification must be provided.

Since the 10 block Gemini data/per frame of the output link are serial in nature only the first block must be identified, the remaining 9 blocks can then be referenced to the first block or sync. The total time between the smallest sub frame is

$$t = \frac{1}{\text{sampling rate}} = \frac{1}{0.416} = 2.404 \text{ seconds}$$

During this sub frame period the total number of input prime frames is:

Number of input prime frames for one input sub frame $= \dfrac{\text{input prime frame rate}}{\text{input sub frame rate}}$

$$= \frac{40 \text{ input prime frame/sec}}{0.416 \text{ input sub frame/sec}}$$

$= 96$ input prime frame/input sub frame

It is only necessary to identify or tag one out of every 10 input prime frame per output frame, the total number of tag words are 96 tag words. The number of bits required to generate the 96 different tag words are 7 bits.

The slowest sub frame of the Agena link is .25 samples per second, therefore identification must be provided. The 4 block of Agena data per frame of the output link are serial in nature, only the first block must be identified. The remaining 3 blocks can be referenced to the first block or sync. The total time between the smalles sub frame is:

$$t = \frac{1}{\text{sampling rate}} = \frac{1}{.25} = 4 \text{ seconds}$$

During this sub frame period the total number of input prime frames is

Number of input prime frame for one input sub frame $= \dfrac{\text{input prime frame rate}}{\text{input sub frame rate}}$

$$= \frac{16}{.25} = 64$$

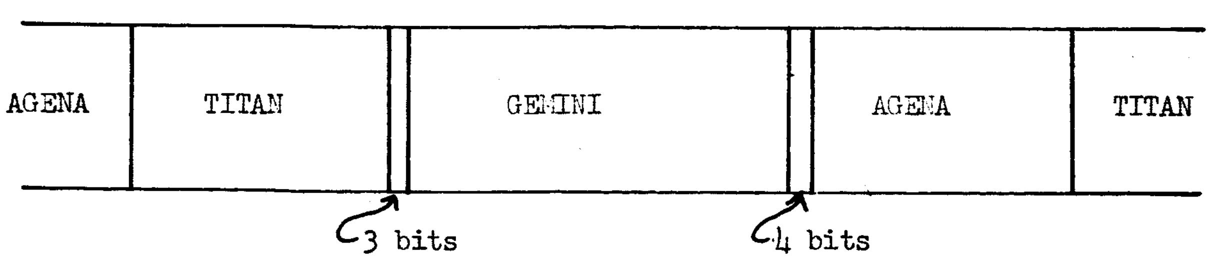
It is only necessary to identify or tag one out of every four input prime frames per output frame. The total number of tag words are 64 tag words. The number of bits required to generate 64 different tag words are 6 bits.

A.5.5 Address Format Conclusions

The total number of tag bits required for the system is the sum of the Gemini and Agena tag bit

$$T_t + T_A = T_{total} = 7 + 6 = 13 \text{ bits}$$

The following diagram shows a typical arrangement of tag bits in the output frame:



A.6 Sync Pattern

The selected sync pattern for the proposed system will be a 21 bit. It is felt that this pattern gives the best probability figures with the minimum number of bits. The total number of words between each prime frame sync of the output channel is 1275. The total bit per output prime output frame is:

$$\text{Total bits} = \text{word}/_{\text{frame}} \times \text{bit}/_{\text{word}}$$

$$= 1275 \times 8 = 10{,}200 \text{ bits}$$

Since the output link will probably not be operating under conditions as severe as the Titan, Gemini, and Agena links, it is felt the optimum compromise between sync time, and bandwidth utilization and assuming a very good bit rate accuracy of the output link, will allow the use of a 21 bit sync pattern for each prime frame of output data or a sync pattern every 10,200 bits.

A.7 Parity and Error Correction

In an output of data which contains 5 Titan, 20 Gemini and 4 Agena input frames, an incorrect data parameter should not be considered critical enough to endanger the mission or cause a serious loss of intelligence. It may not even be necessary to know that a parameter error has occurred during an output frame. However, since it is relatively easy to place an overall parity bit in the output format, a parity bit will be added. It should be noted that the parity checking circuit will only detect the presence of an odd number of errors or an even number of errors, depending on the design of the parity circuit.

More important consideration will be given to time and address data since an uncorrected error in this data could result in the misinterpretation or loss of a complete output frame of data, therefore, not only is it necessary to detect an error but the data bit in error should be corrected.

Consider as an example and address related words containing 'M' significant bits, employing a code which will detect and correct any one bit error of transmittal. This is done by adding 'K' check bits to each word, where the message code including check bits 'K' chosen as follows: (1)

1. Enough 'K' checking bits must be supplied such that the following equation holds true:

   $$2^K \geq M + K + 1$$

   The various values of 'K' necessary for different values of 'M' are:

---

(1) R.H. Hamming, "Error Detecting and Error Correcting Codes", Bell System Technical Journal.

'M'	'K' (Minimum)
1	2
2-4	3
5-11	4
12-26	5
27-57	6

2. The ('M' + 'K') bit positions are numbered from 1 to ('M' + 'K') starting with the least significant bit.

3. The check bits must be chosen such that they serve as parity checks for certain bit positions in the word, the check bits are listed as follows:

Check Bits	Bit Positions
P_0	1,3,5,7,9,11 ....
P_1	2,3,6,7,10,11 1............
P_2	4,5,6,7,12,13,14,15 ....
P_3	8,9,10,11,12,13,14,15,24 ..
P_4	16,17,18,19,20 ....

The general expression for the check bits as a function of bit location is as follows:

Check Bits	Bit Positions
P_n	$(2^n),(2^n+1),(2^n+2),\ldots.(2^n+2^n-1)$
	$(2^{3n}),(2^{3n}+1)\ldots.$etc.

A.8 Sub-Commutation Sampling Rate Compensation and the Utilization of the Output Link as a Function of Parameter Sampling Rates

The Titan PCM data system utilizes 196 analog and 48 bi-level input channels arranged as follows:

Parameters	Rates (samples/sec)
19	400
19	200
36	100
34	40
90	20

The bit rate is 172.8 K b/s and the format is as follows. The prime frame contains 5 sub frames and is sampled 20 times per second. Each sub-frame contains 64 words and each word contains 27 bits and is divided into three 8 bit syllables. Each 8 bit syllable represents one date sample. The 8 bit code is combined in descending order so that the first bit is most significant.

The Titan input link bit rate is faster (172.8 Kb/s than the output link bit rate (40.8 K b/s). The data reduction ratio is therefore determined as follows:

$$\text{Titan Data Reduction Ratio} = \frac{\text{Titan Input Bit Rate}}{\text{Total Output Bit} - (\text{Gemini \& Agena}) \text{ output bit rate}}$$

The minimum Titan data reduction ratio occurs when Gemini and Agena output bit rates are zero and the maximum Titan Data Reduction Ratio occurs when Gemini and Agena output bit rate equal the total output bit rate or in other words when the Titan output bit rate equals zero.

The Titan Data Reduction Ratio is

$$\text{Maximum } DRR_{(T)} = \frac{172 \text{ K b/s}}{40.8 \text{ K b/s} - 40.8 \text{ K b/s}} = \infty$$

$$\text{Minimum } DRR_{(T)} = \frac{172 \text{ K b/s}}{40.8 \text{ K b/s}} = 4.25$$

In other words the data reduction ratio may vary from 4.25 to ∞.

In the following example the minimum Titan data reduction ratio will be considered.

The minimum DRR is not a whole number and thus multiple of the prime rate a direct reduction of Titan parameters cannot be realized, Since full utilization of Titan data cannot be realized the question arises as to just how much utilization can be obtained.

The optimum format of the output link is found by dividing the highest data rate by the Data Reduction Ratio and applying the remainder to the next lowest data rate. Consider the following:

$$\text{Parameter}_{(400)} = \frac{19}{4.25} = 4.5$$

The whole number part of the above equation represents the number of 400 samples per sec parameters contained in the output link, convert the remainder of .5 to the next lowerst sampling rate of 200 samples per second.

$$\text{Parameter}_{(200)(R)} = .5 \times \frac{400}{200} = 1.0$$

adding this to the number of parameters at the 200 samples per second rate and again dividing by Data Reduction Ratio.

$$\text{Parameters}_{(200)} = \frac{19 + 1}{4.25} = \frac{20}{4.25} = 4.7$$

performing similiar operations on the remaining lower rates of 100, 40, and 20, the parameters of all rates can be found. The results are tabulated below:

Parameters	Rates Samples/sec
4	400
4	200
8	100
8	40
21 1	20

The remainder of .3 of the 20 sample per sec rate results in a blank in the output data format every $\frac{1}{.3 \times 20}$ or 1.66 seconds.

The above arrangement may be varied in any combination such that the total number of parameter remain a whole number.

The maximum loss of data for Titan would occur when the remainder of the lowest rate due to various DRR approached 20 as a limit resulting in a blank parameter every .05 seconds.

If the five blocks of Titan in the output link were allowed to vary ±1 parameter from each other then the utilization could be increased by a factor of five (5). This technique will be included when calculating the total utilization efficiency. However, it will not be implemented in the engineering model.

The Titan does not contain sub-frame parameters where as the Gemin and Agena does, therefore, the discussion of sub-frame parameter will be limited to the Agena and Gemini links only.

The Gemini programmer has a master frame and a prime sub-frame. The master-frame consists of 160 words sampled 40 times per second. Higher rates can be obtained by super commutation. However, the discussion will be limited to prime and sub-prime rates. Ninety-six (96) master frames are required to cycle through all data inputs (a requirement based on the prime sub-frame ratios). The prime frame operates four times faster than the sub-frame and every tenth word in the master frame, starting after the sync word, will carry sub-frame data.

The sub-frame consists of 64 words each sampled ten times per second. The sub-frame has further sub-commutation ratios of 8/1 and 24/1 which provides sample rates at 1.25 and .416 samples per second. Twenty-four sub-frames are required to cycle completely all data inputs to this portion of the sub-system.

The output link contains 10 blocks of Gemini data per period or 40 blocks of data per second which is in synchronous with the Gemini prime frame rate of 40 samples per second. If the Gemini link can be totally utilized by the combination of sub-frame parameter, it would seem obvious that the output link having the same prime sampling rate would have total utilization of all output data locations. However, since the input rate is faster than the output rate, the number of parameters would be less by some ratio. The minimum ratio is the ratio of the data rates. Consider the minimum data reduction ratio.

$$\text{Gemini minimum Data Reduction Ration} = \frac{\text{input data rate}}{\text{output data rate}} = \frac{51.2}{40.8} = 1.25$$

Since this number is not a whole number nor is it a multiple of any of the sub-sampling rates, a direct reduction of Gemini parameter cannot be realized. Therefore, full utilization of Gemini data cannot be realized. The question now arises as to just how much utilization can be obtained.

The method for determining the utilization can best be shown by considering a random selection of input parameters.

Parameters	Rate Samples/sec
9	40
5	10
17	1.25
18	.416

The optimum format for the output link is found by dividing the highest data rate by the Data Reduction Ratio of 1.25 and converting the remainder to the next lower rate. The whole number part will be the number of parameter at that rate occurring in the output link, consider first the 9 parameters of the 40 samples per second rate.

$$\text{Parameters}_{(40)} = 9/1.25 = 7.2$$

the whole number part is the number of 40 samples/sec parameter contained in the output link, converting the remainder of .2 to the next lowest sampling rate of 10 samples per second.

$$\text{Parameters}_{(10)(R)} = .2 \times 40/10 = .8 \text{ parameter}$$

adding this to the number of parameters at the 10 samples per second rate and again dividing by 1.25

$$\text{Parameters}_{(10)} = \frac{5 + 8}{1.25} = \frac{5.8}{1.25} = 4.64$$

performing similar operations in the remaining lower rates of 1.25 and .416, the parameter of all Gemini rates can be found. The results are tabulated below:

Parameter	Rates
7	40
4	16
17	1.25
16	.416

The above arrangement may be varied in any combination such that the number of parameters remains a whole number. The maximum loss of data for Gemihi would occur when the remainder of the lowest rate due to various values of DRR approached .416 as a limit resulting in a blank parameter slot in the output format every 2.4 seconds. The loss of data is independent of the number of parameters in the input link. Here again this may be reduced by a factor of ten (10) by varying the 10 Gemini block lengths with an output frame by $\pm$ 1 parameter.

The Agena link has an input data rate less than the output data rate, therefore, the minimum Data Reduction Ratio must be one (1) with blank parameter location filling the unused output location. The maximum Data Reduction Ratio is as in the Titan and Gemini link.

The maximum loss of data for Agena would occur when the remainder of the lowest rate due to various values of D.R.R. approached .25 as a limit resulting in a blank parameter location in the output link every 4 seconds. This may be reduced by a factor of four (4) by varying the four Agena block lengths with an output frame by $\pm$ 1 parameter. The data utilization loss of the three combined links is the total of the Titan, Gemini and Agena Link utilization losses.

$$\text{UL Total} = UL_T + UL_G + UL_A =$$

$$\text{UL (total)} = \frac{20}{1}\ \text{Par/sec} + \frac{1}{2.4}\ \text{Par/sec} + \frac{1}{4}\ \text{Par/Sec} =$$

$$= 20 + .416 + .25 + 20.666\ \text{Par/Sec}$$

$$= 20.666 \times 8 = 165.328\ \text{bit/sec.}$$

by varying the block length by $\pm$ one parameter, the following utilization loss could be realized.

$$\text{UL (total)} = \frac{20}{5} + \frac{.416}{10} + \frac{.25}{4} = 4 + .0416 + .0625$$

$$= 4.1041\ \text{parameter/sec or } 32.8328\ \text{bits/sec.}$$

or rounded off to 9 bits/output frame

A.9 Bandwidth Utilization Efficiency

Bandwidth Utilization is defined as the number of bits available for the transmission of data parameters as compared to the total number of bits. The bits which represent the difference between the total number of bits and the bit available for data transmission are used for correlating time, sub-frame identification, bit rate tolerance compensation, sync, error identification and correction, and sampling rate compensation. The total bits required for time, sub-frame identification, sync and error identification and correction which an output link bit format requires are listed. The remaining bits are listed below:

For the Titan, Gemini and Agena case:

format -	297 bits
Data Overflow Compensation	11 bits
Sampling Rate Compensation	9 bits
	317 bits

Converting the non-data bits to words thus:

$$\text{non-data words} = \frac{317}{8} = 39\ \frac{5}{8} \text{ or } 40 \text{ words/output frame}$$

the bandwidth utilization efficiency is determined as follows:

$$\text{B.U.E. in \%} = \frac{1275-40}{1275} \times 100 = 96.2\%$$

A.10 Extrapolation to a General Requirement

Previous discussion has centered on 3 known data channels (Titan, Gemini and Agena) however, consideration must be given to multi-channel, multi-rate and synchronous systems. Consider five high prime number frame rate input channels having frame rates A, B, C, D, and E and an output channel having a data rate F. It has been previously stated that the period of the output channel (i.e., sync separation time) is dependent on the ability to reconstruct each data parameter from the signal. The maximum separation between sync pattern recommended by experimenters is approximately 10,000 data bits. The sync rate of the output link is, therefore

$$S_{RO} = \text{Output Sync Rate} = \frac{\text{Output bit rate}}{10{,}000}$$

If we also assume that the prime frame rate is equal to the prime sync rate, then

$$F_{ro} = \text{Output frame rate} = S_{po} = \frac{\text{Output bit rate}}{10{,}000}$$

In order to extract data from the output link, it is necessary that over some period of time, F_{ro} must equal a whole number.

$$F_{ro} = \frac{F}{N}$$

where N is some number less than 10,000 which allows F_{ro} to equal the minimum possible whole number and F and f_{ro} are expressed as per period rates, for simplicity the period is chosen as one second.

In order to convert the input-output systems from asynchronous to synchronous, it is necessary that the input frame rates of each channel be divided by F_{ro} be a whole number. If the number is not a whole number, then the effective rate must be increased until it is a whole number. The whole number represents the number of frames of the first channel which appears in each frame of the output channel.

$$F_A = \text{Input Frames/Output Frame} = \frac{A.\ P + (fro - N_A)}{F_{ro}}$$

where A - is the frame rate of the input channel in frames per second

P - is in seconds/period

N - is any whole number between 1 and f_{ro} which makes R the smallest possible whole number.

FUNCTION	M BITS	K BITS	TOTAL
SYNC	21	-	21
TITAN TIME OF DAY (D_1)	34	6	40
TITAN SYNC TOLERANCE (D_2)	6	4	10
TITAN SYNC TOLERANCE (D_3)	7	4	11
TITAN SYNC TOLERANCE (D_4)	8	4	12
TITAN SYNC TOLERANCE (D_5)	8	4	12
GEMINI CHANNEL SEPARATION AND ADDRESS	21	5	26
GEMINI SYNC TOL. (D_2')	5	4	9
GEMINI SYNC TOL. (D_3')	6	4	10
GEMINI SYNC TOL. (D_4')	7	4	11
GEMINI SYNC TOL. (D_5')	7	4	11
GEMINI SYNC TOL. (D_6')	7	4	11
GEMINI SYNC TOL. (D_7')	8	4	12
GEMINI SYNC TOL. (D_8')	8	4	12
GEMINI SYNC TOL. (D_9')	8	4	12
GEMINI SYNC TOL. (D_{10}')	8	4	12
AGENA CHANNEL SEPARATION & ADDRESS	22	5	27
AGENA CHANNEL SYNC TOL. (D_2'')	6	4	10
AGENA CHANNEL SYNC TOL. (D_3'')	7	4	11
AGENA CHANNEL SYNC TOL. (D_4'')	8	4	12
GUARD & PARITY	2	3	5

TOTAL SYNC, TIME, ADDRESS, ERROR CORRECTION AND PARITY BITS FOR ONE OUTPUT FRAME IS:

297 Bits

Similarly:

$$F_B = \frac{B \cdot P + (F_{ro} = N_B)}{F_{ro}}$$

$$F_C = \frac{C \cdot P + (F_{ro} - N_o)}{F_{ro}}$$

$$F_D = \frac{D \cdot P + (F_{ro} - N_p)}{f_{ro}}$$

$$F_E = \frac{E \cdot P + (f_{ro} - N_E)}{f_{ro}}$$

therefore, the format for one output frame shall contain X input frames where

$$X = R_A + R_B + R_C + R_D + R_E$$

the effective frame utilization is

$$F.U. = \frac{(A + B + C + D + E)\ P}{(A + B + C + D + E)\ P} \qquad N = E\ (f_{ro} - N$$

$$\frac{(A + B + C + D + E)\ P}{(A + B + C + D + E)\ P + K\ f_{ro} - (N_A + N_B + N_C + N_D + N_E)} \qquad N = A$$

where K is the number of input channels.

Consider the frame rates for three known input links, Titan, Gemini and Agena and a 40.8 Kb/sec output link

$$S_{RO} = \frac{40,800}{10,000} \approx 4$$

$$F_{ro} = 4$$

$$R_A(\text{Titan}) = \frac{A \cdot P + (f_{ro} - N_A)}{f_{ro}}$$

where $P = 1$, $f_{ro} = 4$, and $A = 20$

$R_A = \frac{20 \times 1 + 4 - N_A)}{4} = \frac{20}{4} = 5$ where $N_A = 4$

$R_B = (\text{Gemini}) = \frac{40}{4} = 10$ where $B = 40$ and $N_B = 4$

RC (agena) $= 16/4 =$ where $C = 16$ and $N_C = 4$

$$F.U. = \frac{(A + B + C)P}{(A + B + C)\ P + (f_{ro} - N)} \quad N = C \quad N = A$$

since NC

$$\sum_{NA}^{NC} (f_{ro} - N) = 0$$

then F.U. = 100%

Consider a second example where the input frame rates are 13, 20 and 28 samples/sec and the output data rate is 28,000 bits/sec.

$$f_{ro} = \frac{F}{N} = \frac{28{,}000}{9{,}333} = 3$$

$$R_A = \frac{13 + (3 - NA)}{3} = 5 \text{ where } N_A = 1$$

$$R_B = \frac{20 + (3 - NB)}{3} = 7 \text{ where } N_B = 2$$

$$R_C = \frac{28 + (3 - NC)}{3} = 10 \text{ where } N_C = 2$$

$$F.U. \ \frac{13 + 20 + 28}{13 + 2- + 28 + (2+1+1)} = \frac{61}{65} = .94$$

The ratio of 61/65 can be stated that for every 65 input frames/output frames only 61 will represent true input frames and 4 will contain no data.

The effective sync separation utilization must also be considered. It was noted that the optimum sync separation must be reduced in order that F_{ro} is a prime number so that synchronization can be obtained. The sync separation utilization is therefore

$$S.U. = \frac{\text{Sync Separation}}{\text{Max. Sync Separation allowable}}$$

The S.U. for the first case is 100% and for the second case is

$$S.U. = \frac{9{,}333}{10{,}000} = .93$$

In order to relate the frame and sync utilization to the overall system, it is necessary to convert them to a bit and then to a bandwidth utilization factor. The effect sync bit utilization loss is determined as follows:

$$S.B.U.L. = \text{sync bits} \times (1 - S.U.)$$

for the first case

$$S.B.U.L. = 21 \times (1.0 - 1.0) = 0$$

for the second case

$$S.B.U.L. = 21 \times (1 - .93) = 1.47 \text{ or } 2 \text{ bits/frame}$$

The effect of frame bit utilization loss effect on B.U.E. is more serious.

$$F.B.U.L. = N \times (1 - F.U.)$$

For the first case

$$F.B.U.L. = N(1-1) = 0$$

For the second case

$$F.B.U.L. = 9{,}333 \times (1-.94) = 560 \text{ bits}$$

The decrease in bandwidth utilization efficiency, due to frame and sync utilization less than unity, is determined as follows:

$$\text{Non-Data Bits} = F.B.U.L. + S.B.U.L.$$

$$B.U.E. \text{ (decrease) in percent} = \frac{F.B.U.L.+S.B.U.L.}{\text{Bits/Frame}} = 100$$

for the case of three input channels with frame rates of 13, 20, and 28 samples/sec and an output channel with a data rate of 28,000 bits/sec.

$$B.U.E. \text{(decrease) in percent} = \frac{562}{9{,}333} \times 100 = 6\%$$

A.11 Design Criteria for the Block Burst Retransmission System

The optimum design and evaluation of the Block Burst Retransmission System is based on the characteristics of the input and output channels which enter and leave the retransmission system. The characteristics are usually given and the system must then be designed and optimized around the characteristic parameters. This section discusses the effect of variable multi-channel input parameters on the retransmission system without restrictions on any of these parameters. The following discussion will give step-by-step the criteria which must be specified for the design of a blocked burst retransmission system, and in addition the discussion establishes the basic equations for determining the limitations of the system as a function of the given parameters. System design criteria are related to the selection of the input given and output parameters which must be optimized are listed as follows:

1. Period of output frame.
2. Input frames per output frame.
3. Address bits.
4. Data ratetolerance bits.
5. Channel Separation bits.
6. Time bits.
7. Error Correction and Parity check.
8. Memory capacity.
9. Data over flow compensation.
10. Sub-commutation compensation.
11. Bandwidth efficiency.

Retransmission systems in which the total input data rate is greater than the output data rate and in which data compaction techniques are not used, requires the pre-selection of data parameters and their locations, conversely, the data parameter not selected and their locations are then also known. Location of non-selected parameters between the first selected parameter and the last selected parameter are referred to as holes and will be used in determining memory storage requirements.

The more important parameter which must be specified in order to fully evaluate the retransmission system are listed in the following table. It is assumed that for any particular system evaluation, the parameters are held constant.

Specified Parameters of the Retransmission System

Channel	Data Rates	Prime Frame Rates	Selected Words	Selected Holes	Sampl.Rate
Input 1	R_1	F_1	W_1	H_1	$S_1S_1S_1$..
" 2	R_2	F_2	W_2	H_2	$S_2S_2S_2$...
" 3	R_3	F_3	W_3	H_3	$S_3S_3S_3$...
Input N	R_n	R_n	W_n	H_n	$S_nS_nS_n$...
Output	R_o	To be calculated	W_o	NA	

A.11.1 General Output Word Format

The retransmission output link contains the data parameter and the time of occurrence of the parameter. In order to provide this information with the optimum bandwidth utilization, it is necessary to devise a special format for the retransmission systems output. The output format for retransmitting input channel number, one contains data parameter words grouped as to prime frame number, an output time word and sync deviation word. The time and deviation words are used to locate the prime frames with respect to time. The actual parameter selection routine of the programmer determines the location of the parameter within the frame. This information must also be available at the data reduction site to properly locate the selected parameter with respect to time. In addition address (frame) identification is provided at the beginning of the block of frames of the channel. The format for the remaining input channels is similar to the first except they contain channel seperation words instead of a time of day word. The time of occurrence for the first frame of these input channels is determined by summing the time of day word of the first channel with the channel separation word of the channel under evaluation.

A.11.2 Period of Output Frame

It has been previously stated the period of the output channel (i.e., output sync separation time) is dependent on the ability to reconstruct each data parameter, therefore, from experimental data

$$S_o\text{(output sync rate)} = f_o\text{(output frame rate)} = R_o/K$$

$$\text{or } T_p\text{(period)} = K/R_o$$

where K is any whole number less than 10,000 and divisible by 8 which allows f_o to e qual the minimum possible whole number and where R_o and f_o are expressed in per period rates.

The separation between sync for the ideal system is R_o/N where N=10,000, however, in order to synchronize the asynchronous input channels it is necessary to reduce the value of N. This reduction prevents the full utilization of sync separation allowable, therefore, a sync utilization loss results. This loss is determined as follows:

$$\text{Sync Separation Loss } (\alpha k) = \left(1.9 - \frac{\text{Sync Separation}}{\text{Max.Sync.Separ.Allow.}}\right)$$

$$= \left(1.0 - \frac{N}{10,000}\right)$$

The sync utilization efficiency in percent (ηK) can be determined from αK.

A.11.3 Input Frames per Output Frames

Synchronous timing periods are necessary if a time of occurrence data is not sent for every parameter. In order to convert the retransmission system from asynchronous to synchronous, it is necessary that the input frame rates of each channel divided by F_o equal a whole number. If the number is not a whole number then the frame rate must be increased by introducing blank frames. until it is a whole number. The resulting whole number represents the number of frames of the input channel plus blank frames which appear in each frame of the output channel. The difference between the initial rate and the new rate divided by the new rate is the frame utilization loss. The frame utilization loss subtracted from unity is the frame utilization factor.

The frame rate required for synchronizing the input channels to the output channel is determined as follows:

$$A_N = \text{input frames/output frame}$$

$$= \frac{F_N + (F_o - B_N)}{F_o}$$

where F_N is the frame rate of the input channel

B_N is any whole number between 1 and F_o which makes A_N the smallest possible whole number.

The new effective frame rate for synchronization of the input channel to the output channel is F_N' and is determined as follows:

$$F_N' = F_N + (F_o - B_N)$$

The frame utilization loss is determined as follows:

$$\alpha_f = \frac{\sum_{n=1}^{n=} F_N' - \sum_{1}^{\infty} F_N}{\sum_{1}^{\infty} F_N'} = \frac{\sum_{n=1}^{n=\infty}\left[F_N + F_o - B_N\right] - \sum_{n=1}^{n=\rho} F_N}{\sum_{n=1}^{n=\infty}\left[F_N + F_o - B_N\right]}$$

$$= 1 - \frac{\sum_{n=1}^{n=\infty} F_N}{\sum_{n=1}^{n=\infty}\left[F_N + F_o - B_N\right]} = 1 - \sum_{n=1}^{n=\infty} \frac{F_N}{F_N + F_o - B_N}$$

The frame utilization efficiency is percent (η_f) can be determined from α_f

$$\eta_f = \left[\sum_{n=1}^{n=\rho} \frac{F_N}{F_N + F_o - B_N}\right] 100$$

$$= (1 - \alpha_F)\ 100$$

The system designer should exercise care wherever possible in the selection of the input and output channel parameters in a multi-channel system. The selections should be made to optimize the sync and frame utilization efficiency.

A.11.4 Address (Frame Identification)

The basic requirement of an asynchronous system is to be able to determine which data parameter is being sent as well as its time of occurrence. The identification of an input parameter in the output channel requires some form of address or identification in the output format. If a parameter occurs at a prime or super commutation rate, it is only necessary to provide time of sync as a means for determining time of sync. For channels which have sub-commutated data rates, it is necessary to determine which prime (main) frame contains the sub-commutated parameter. In order to locate the sub-commutated parameters, it is necessary to place an identification word at the beginning of each block of output data from a channel. The word would identify only the first frame of the blocks. Since the following frames occur in arithmetic progression identification could be obtained by referencing the first frame.

The number of input prime frames for one input sub-frame is:

$$P_N = \frac{\text{input prime frame rate}}{\text{lowest input sub-prime rate}}$$

The number of bits (I_o) required for address (frame) identification is determined as follows:

$$(Io)A = P_N$$

$$(I_o)A \log_{10} 2 = \log_{10} P_N$$

$$(I_o)A = \frac{\log P_n}{\log 2}$$ where (Io)A is the closest larger whole number.

The bandwidth utilization efficiency in percent ηA due to the address word is:

$$\eta A = \frac{K - (Io_1 + Io_2 + Io_3 + \ldots I_{oN})\ A}{K} \times 100$$

$$\eta A = (1 - (Io_1 + I_{o2} + Io_3 \ldots I_{oN})\ A \times 100$$

The bandwidth utilization loss ($\propto A$) due to the address can be determined in terms of ηA.

$$\propto A = \sum_{N=1}^{N=} \frac{1}{K} \left| \frac{\text{Log } P_N}{\text{Log } 2} \right|$$

$$\eta A = 100\ (1 - \propto A)$$

$$\propto A = 1 - \frac{\eta A}{100}$$

The effect of $\propto A$ will be small when compared to other losses, however, the system designer should consider sub-prime commutation rates which cause I_o to be equal to or slightly less than a whole number for if I_o is slightly larger than a whole number then the next larger whole number must be used.

A.11.5 Data Rate Tolerance

Once time is established at the start of a block of frames of input channel "N" it is only necessary to determine the deviation in the known input data rate in order to determine the "time of occurrence" for any input data parameter. The deviation can be the result of frequency drift, doppler shift or other similar causes. The main cause of deviation, however, is due to frequency drift of the bit rate oscillator of the data gathering system. The maximum time deviation per frame is determined as follows:

Deviation(max.)/frame = Data rate tolerance x prime frame rate.

In order to determine the deviation in time between the time or channel separation word and a particular frame (N) of the channel it is only necessary to multiply the deviation per frame by the number of frames (N-1) between the time or channel separation word and the frame in question.

Total deviation (max) = Data rate tolerance x prime frame rate x (N-1) or frame (N)

The maximum time deviation for a given frame is formulated into a deviation word and placed at the beginning of each frame. The following table is the time deviation for n frames.

Frame No.	Deviation Word	Deviation Word Number
Frame #1	Time of Day (Channel Seperation)	#1
" 2	Deviation/frame x 1	#2
" 3	" x 2	#3
Frame N	Deviation/frame x (N-1)	#N

The number of bit $(I_o)t$ required for the time deviation word is determined as follows:

$$2^{I_o} = \text{Deviation word}$$

$$(I_o)\, t = \frac{\log\ (\text{deviation word})}{\log\ 2}$$ Where I_o is the closest larger whole number.

The normal amount of bits required for deviation words are extremely small when compared to the bits required for time of day words, however, since the deviation words is directly related to the bit rate tolerance and the number of frames per block, it is essential that the system designer strive to keep the input rate tolerance tight and the prime frame rate to some common multiple of a low prime number.

The bandwidth utilization efficiency in percent Nt due to bit rate tolerance is:

$$\eta_T = \frac{K - (I_{o_2} + I_{o_3} + I_{o_4} + \ldots\ldots I_{o_N})_T}{K} \times 100$$

where $(I_o)_T$ represent the deviation word bits.

$$\eta_T = 100 \left[1 - \frac{(I_{o2} + I_{o3} + I_{o4} + \ldots\ I_{oN})_T}{K}\right] = \left[1 - \frac{1}{K} \sum_{n=2}^{n=\infty} \left|\frac{\mathrm{Log}\ D_N}{\mathrm{Log}\ 2}\right|\right] 100$$

The bandwidth utilization loss (αT) due to bit rate tolerance can be determined in terms of $\eta\, T$

$$\eta_T = \sum_{n=2}^{n=\infty} \frac{\mathrm{Log}(\mathrm{Dev})N}{K\ \mathrm{Log}\ 2}$$

$$\eta_T = (1 - \alpha_T)\ 100$$

$$\alpha_T = 1 - \frac{\eta_T}{100}$$

A.11.6 Channel Separation Word

The previous discussion centered on establishing time of occurrence for an individual frame within the first input channel using the time of day word at the beginning of the output frame and the bit rate tolerances of the input channel. Once time of occurrence for the beginning of each successive block is established then the frame timing can be accomplished in similar manner. The time of occurrence of a block can be determined by summing the time of day word and the time separation between the first block and the data block under consideration. The maximum time separation is the period of the output frame where

$$T_p(\text{period}) = \frac{K}{R_o}$$

The channel separation word must have the capability of handling the max. time separation (T_p).

The max. bits required for channel separation word is determined as follows:

$$2^{(I_o)\,S} = \text{Separation Time Word } (T_p)$$

$(I_o)_S = \dfrac{\log T_p}{\log 2}$ where Io is the closest larger whole number and where n represents the block number.

The bandwidth utilization efficiency is percent η_s due to the channel separation word is:

$$\eta_s = \frac{K - (I_{o2} + I_{o3} + \ldots\ldots I_{o4})_s}{K} \times 100$$

$$= \left[1 - \frac{(I_{o2} + I_{o3} + \ldots\ldots I_{oN})s}{K}\right] \times 100$$

$$= 100\left(1 - \sum_{n=2}^{n=\infty} \frac{I_{oN}}{K}\right)_s = 100\left(1 - \sum_{n=2}^{n=\infty} \frac{N}{K} \left|\frac{\text{Log } T_p}{\text{Log } 2}\right|\right)$$

The bandwidth utilization loss ( ) due to the address can be determined in terms of

$$\alpha_s = \sum_{n=2}^{n=\infty} \frac{N}{K} \left|\frac{\text{Log } T_p}{\text{Log } 2}\right|$$

$$\eta_s = 100(1 - \alpha_s)$$

$$\alpha_s = 1 - \frac{\eta_s}{100}$$

A.11.7 Time Word

The time of occurrence of the first sync within the period is detected and used to generate the time word. The time word will contain units of hours, minutes and second and will be measured to an accuracy of one microsecond. The number of bits required for the complete time word is tabulated as follows:

Number System	Bit Location			Total Bits
	Hours	Min.	Sec.	
1) Binary coded decimal	6	7	31	44
2) Straight Binary with separation between units of hrs. Min., and Sec.	5	6	26	37
3) Straight binary with no separation				37

The number of bits listed above are based on a maximum number of 24 hours, 60 minutes, and 60 seconds.

Since the primary interest is in transmitting the maximum data within a period rather than standardizing the time format, only the 2nd system which uses the least number of output data bits yet has some form of time separation will be considered.

The bandwidth utilization efficiency in percent η_t due to the time word is:

$$\eta_t = \frac{K - T_L}{K} 100 \qquad T_L = \text{Time Word}$$

The bandwidth utilization loss α_t due to the address can be determined in terms of η_t :

$$\alpha_t = \frac{T_L}{K}$$

$$\alpha_t = 1 - \frac{\eta_t}{100}$$

A.11.8 Error Correction and Parity Check

An incorrect data parameter should not be considered critical enough to endanger the mission or cause a serious loss of intelligence. It may not even be necessary to know that a parameter error has occurred during an output frame, however, since it is relatively easy to place an overall parity bit in the output format, a parity bit is added.

More important consideration is given to time and address data since an uncorrected error in this data could result in the mininterpretation or loss of a complete output frame of data, therefore, not only is it necessary to detect an error but the bit in error should be corrected.

A single bit error can be corrected using an error correction code containing "K" checking bits. The number of "K" bits for "N" bits of data are listed in the following table.

Data Bits	Correction
1	2
2-4	3
5-11	4
12-26	5
27-57	6
"	"
"	"
'N'	'K'

The correction bits. when considering bandwidth utilization efficiency or loss are included as a part of the data words.

A.11.9 Memory Capacity

The total storage required 'n' input channels not considering overlap of memory access between input and output, is determined as follows:

$$M_T = \sum_{n=1}^{n=\infty} T_n' \; S_n \, W_n + \frac{T_n'' R_n}{8} + \left(W_n - \frac{T_n'' R_n}{8}\right) \frac{R_n - R_o}{R_n}$$

where $\left[W_n - \frac{T_n'' R_n}{8}\right] \geq 0,$

and $\frac{T_n'' R_n}{8} + \left(W_n - \frac{T_n'' R_n}{8}\right) \frac{R_n - R_o}{R_n} \leq W_n$

T_n' is the whole number part of T_n and $T_n'' = (T_n - T_n')$

$$T_n = \frac{N}{R_o} + 8 \left[W_n\left(\frac{1}{R_o} - \frac{1}{R_n}\right) - \frac{H_n}{R_n}\right] + \frac{1}{S_n} + \frac{8 W_n S_n K}{R_o^2}$$

The definition of the parameters are as follows:

S_n - Sync rate of input channel n

W_n - Words per frame of input channel n

R_n - date rate of input channel n

R_o - data rate of output channel

H_n - word gaps between first and last selected data parameter of a frame of channel n.

The time parameters T_n' and T_n'' are functions of T_n and are defined as follows:

$$T_n = T_n' + T_n''$$

Where T_n is the time between the start of data storage of channel n and the start of data read out. T_n' is the whole number part of T_n while T_n'' is the remainder.

The parameter T_n is determined from the following equation

$$T_n = T_p + T_x + T_{(slip)} - T_t$$

where T_p is the period of time $\frac{1}{f_o}$ or $\frac{K}{R_o}$ and, T_x is the receiver time less the transmit time of one frame of channel n. $T_{(slip)}$ is the input frame granularity $\frac{1}{S_n}$ and T_t is the output time of input data channel n.

$$T_x = 8\left[W_n\left(\frac{1}{R_o} - \frac{1}{R_n}\right) - \frac{H_n}{R_n}\right]$$

$$T_t = \frac{8}{R_o} W_n \frac{S_n}{R_o} K = \frac{8\ W_m S_n K}{R_n^2}$$

combining and solving for t_n

$$T_n = \frac{N}{R_o} + 8\left[W_n\left(\frac{1}{R_o} - \frac{1}{R_n}\right) - \frac{H_n}{R_n}\right] + \frac{1}{S_n} + \frac{8\ W_n S_n K}{R_o^2}$$

The maximum data loss due to data overflow compensation is determined as follows:

Data loss/output frame:

$$\sum_{n=1}^{n=\infty} \frac{\left(R_{n(max)} - R_{n(min)}\right) R_{o(min)} W_n}{F_o R_{n(max)} W_o}$$

where R_n (max) is the max. input data rate for channel n

R_n (min) is the min. input data rate for channel n

R_o (max) is the max. output data rate

F_o is the output frames per second

W_n is the number of words of channel 'n' per output frame

W_o is the total number of words per output frame

The data utilization loss α_D is determined as follows:

$$\alpha_D = \frac{\text{Data loss/output frame}}{\text{Total Data/output frame}} = \frac{\text{Data loss/output frame}}{8W_o}$$

$$\alpha_D = \sum_{n=1}^{n=} \left[\frac{R_n(max) - R_n(min) \qquad R_o(min) \quad W_n}{8 \quad F_o \quad R_n(max) \qquad W_o^2} \right]$$

A.11.11 Sub-commutation Compensation

The loss due to sub-commutation compensation can be controlled by the proper selection of parameters and data rates. The general PCM data system when not considering parameters priorities is arranged as follows:

Input Parameters	Rates (samples/sec.)
X_n	S_n
X_n'	S_n'
X_n''	S_n''
X_n'''	S_n'''
etc.	$S_{n(lowest)}$

The Data Reduction Ratio is determined as follows:

$$DRR = \frac{\text{Channel n input bit rate}}{\text{Output bit rate - output bit rate for all channels other than n}}$$

$$\text{Max. DRR} = \frac{R_n}{R_o - R_o \dfrac{W_o - W_n}{W_o}} \longrightarrow \infty$$

as $W_n \rightarrow 0$

$$\text{Min. DRR} = \frac{R_n}{R_o - R_o \dfrac{W_o - W_n}{W_o}} \longrightarrow \frac{R_n}{R_o}$$

as $W_n \longrightarrow W_o$

Consider the following:

$$\text{Parameter}(S_n) = \frac{X_n}{\text{DDR}} = Y \; Y_n$$

The whole number part of Y represents the number of S(n) samples per seconds parameters contained in the output link, converting the remainder to the next lowest sampling rate.

$$\text{Parameter } (S_n') \text{ (remainder)} = Y_n - \frac{S_n}{S_n'} = R'$$

adding this number to S_n' and again dividing by the DRR

$$\text{Parameter } S_n' = \frac{X_n' + R'}{\text{DRR}} = Y' \; Y_n'$$

again the whole number part Y' represents the number of S_n' samples per sec. parameters contained in the output link. The above steps are repeated until all parameters rates are accounted for.

The remainder of the last conversion represents the utilization loss due to sub-prime sampling rates. The maximum parameters loss for channel n would occur when the remainder of the lowest rate equals the magnitude of the lowest rate as a final value.

Therefore, the utilization loss in bits per frame is determined as follows:

$$UL = \sum_{n=1}^{n=} \frac{S_n \text{ (lowest)}}{F_o} \times 8$$

$$\alpha_R \quad \sum_{n=1}^{n=\infty} \frac{S_n \text{ (lowest)}}{F_o W_o}$$

A.11.02 <u>Bandwidth Utilization Efficiency</u>

The total bandwidth utilization loss is determined by summing the utilization loss due to formatting and compensating factors, determined in previous paragraphs.

$$\alpha \text{ BUL (total)} = \alpha_S + \alpha_F + \alpha_A + \alpha_T + \alpha_C + \alpha_t + \alpha_E + \alpha_D + \alpha_R$$

where α_S is the sync synchronous loss

α_F is the frame utilization loss

α_A is the address utilization loss

α_T is the rate tolerance utilization loss

α_C is the channel separation

α_t is the time word utilization loss

α_E is the error correction utilization loss

α_D is the data overflow compensating utilization loss

α_R is the sub-commutation compensation utilization loss

The total bandwidth efficiency is determined as follows:

$$\text{(bandwidth)(total)} = (1 - {}_{\text{BUL(total)}}) \ 100$$

Burst Data Reformatting Program

List of Illustrations

Tables

B-1. GENERAL:

This document prescribes the requirements and specifications for a computer program which will demonstrate the feasibility of the Periodic Burst Technique.

B-2. PURPOSE:

The purpose of the study was to examine a technique for retransmitting telemetry data taken from three different input links. This program will test this technique by processing simulated data from three tape stations, sampling and reformatting that data via the predetermined format routine, and outputting to a fourth tape drive.

B-3. PROGRAM PHILOSOPHY:

B-3.1 Discussion

The input data will be taken from each of three tapes transports which will hold data recordings similar to transmissions from Titan, Gemini, and Agena telemetry links.

The different data rates from each of the missiles, 172.8 kilobits per second for Titan, 51.2 kilobits per second for Gemini, and 16.384 kilobits per second for Agena will be scaled so that the data rate for Titan is three times that of Gemini, and that for Agena is one-third of Gemini.

Table B-1

Vehicle	Real Data Rate-KBPS	Scaled Data Rate
Titan	172.8	172.8
Gemini	51.2	51.2(ref)
Agena	16.384	17.07

Vehicle Data Rates

The actual data input rate will be about one-quarter that of the scaled data rate since the peak input data rate is limited by the computer to 41.7 kilobits per second.

The input data will have been addressed prior to introduction to the computer as if it was the output of the PCM decommutator. The data will then be reformatted into the format dictated by the program.

Output will be on tape.

A variation of the internal timing rate will be possible.

B-3.2 Advantages

The program will demonstrate the capability of the routine of selecting parameters from three different sources running asynchronously, and retransmitting them by the periodic burst technique.

The feasibility of varying the input rate and the ability to reconstruct a time base will be demonstrated.

B-3.3 Exclusions

The primary limitation in demonstrating a real-time data handling system is the simulation of the variable, ±0.05% on Titan, data input rate. Inasmuch as the input tape recording cannot be varied at will (and if the input rate could be varied, the input buffer would restore the data to the computer determined rate), an artifice of defining everything in times of periods (nominally of 102 bits duration) and varying the period by ±1%), will demonstrate the usefulness of the guard bit.

A second problem occurs in meeting the input data rates. The input rates cannot be simulated exactly due to the limitation on the tape input speed. Therefore, a read-in rate one-eighth of the maximum (Titan) rate resulted. The output rate closely simulates the required rate for the communications link although the figures are circumstantial at this point.

The data processing rate for the computer is certainly within the telemetry input specification since the basic transfer rate is on the order of 2 microseconds (500 Kbits per second).

The simulation of the various data rates has been approached closely by varying the ratios of useful to total data (input) by the ratios 9:3:1.

Real telemetry data has not been used primarily because the actual computer input assumes that the PCM data has been decommutated and addressed. Also, a number of sub-commutated channels have been dropped since they add little to the reformatting problem.

The input to many of the channels has been held constant to simply the analysis of the results.

In using a number of tape decks which are part of a working system, the bit-by-bit variation between input channels which would result using three actual telemetry sensors will not occur. The very syncing of the tape drives by the computer will tend to make the input data less variable than in the final system.

B-4. INPUT DATA:

B-4.1 Tape Recorder Data

B-4.1.1 General

Tape Recorder Data will be from three tape stations, simulating the Titan, Agena, and Gemini missiles. The ratio of data words will be 9 Titan words to 3 Gemini words to 1 Agena word.

By scaling upwards:

(a) 3 Gemini words/unit time----51.2 KBPS
(b) 9 Titan words/unit time----153.6 KBPS
172.8 KBPS (incl.sync)
(c) 1 Agena word/unit time----- 17.1 KBPS
as compared to 16.384 KBPS in reality.

B-4.1.2 Data Format

Data will be a format similar to that which would emerge from the decommutator (after addressing). Data will be in 36 bit standard binary format. Tape should be in high density.

B-4.1.3 General Word Format

The general format will be the same for all data words:

Bit position # 1-9	First channel address
10-18	First data word
19-27	Second channel address
28-36	Second Data word

B-4.1.4 Specific Word Identification

Data which is transcribed on each of the tapes will follow the patterns listed below. For purposes of simplicity the mechanical analog of the format, the sub-commutation format, has been used.

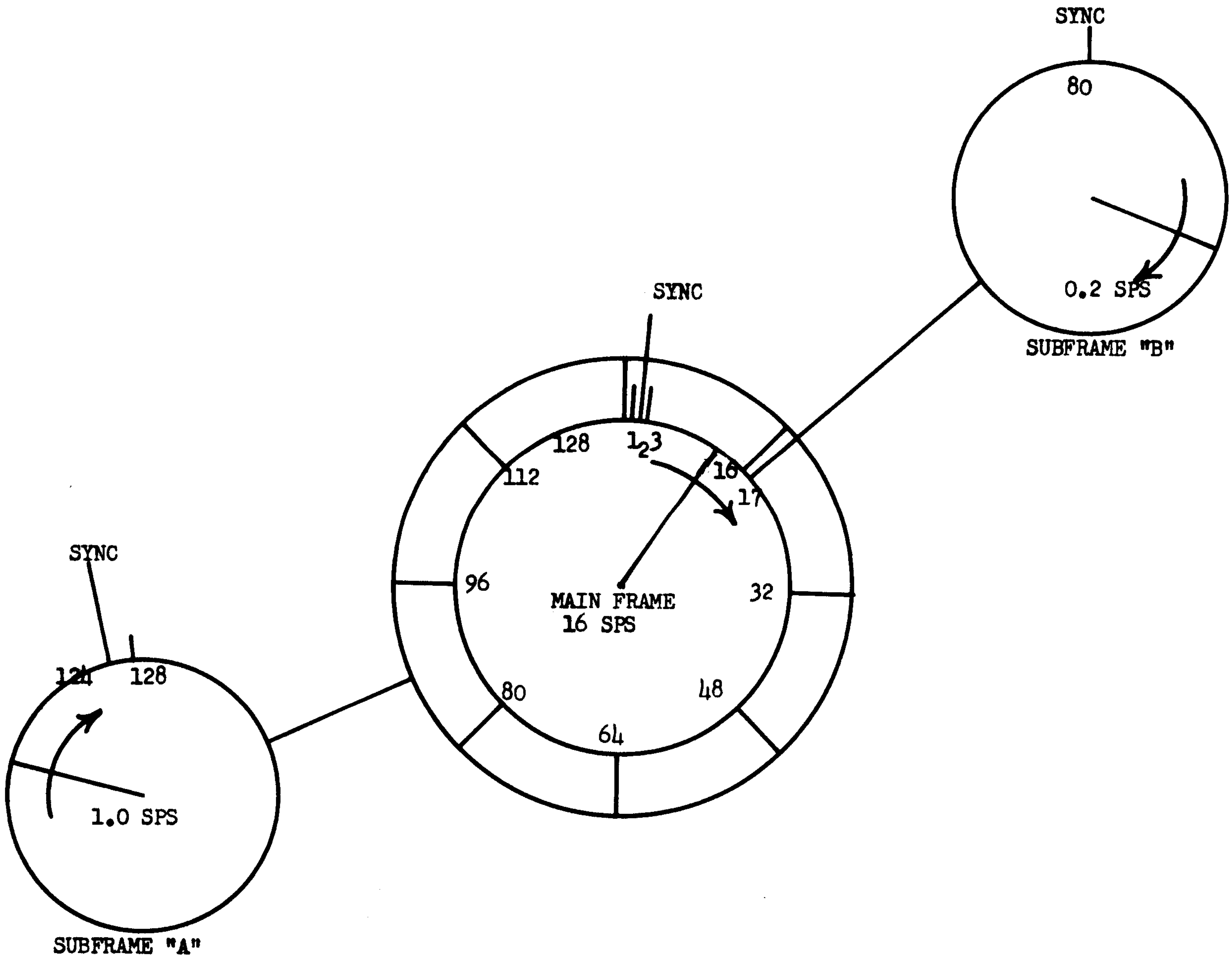


FIG. B-1 AGENA COMMUTATION ANALOG SIMPLIFIED FOR PERIODIC BURST DEMONSTRATION

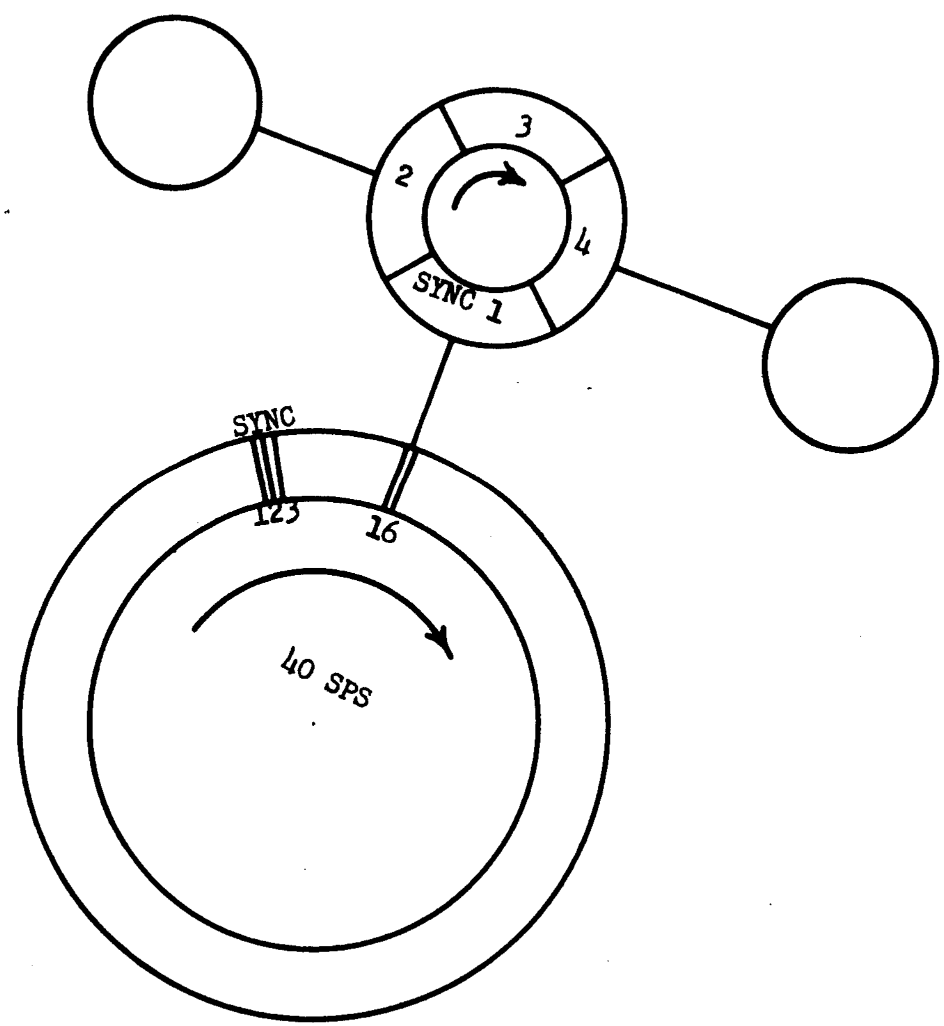


FIG. B-2 GEMINI COMMUTATION ANALOG, SIMPLIFIED FOR PERIODIC BURST DEMONSTRATION

The format for making up the input data tapes should be capable of allowing different simulated data information to be substituted into some of the data channels.

B.4.1.4.1 Agena Data

B.4.1.4.1.1 Simplification

For purposes of the demonstration, the Agena sub-commutation format will be simplified (see figure B-1). Sub-frame C and the last two parts of sub-frame B will be dropped.

B.4.1.4.1.2 Specific Channel Contents

Main channels 4-128, with the exception of 16, 17, 32, 48, 64, 80, 96, 112, and 124, should contain the channel number itself. The same will hold for sub-frame A. The excluded channels are those which lead to sub-commutated rates. Sub-channel B should follow the same assignment pattern. Channel 1 contains sync which is shown in Figure 3.2.17 entitled Agena Telemetry, PCM System Synchronization Words.* Note that all commutation rates are met.

B.4.1.4.2 Gemini Data

B.4.1.4.2.1 Simplification

All but one of the sub-commutated pick-offs has been eliminated. This is shown in Figure B-2, Simplified Commutation Format Gemini Telemetry. As in the case of Agena, at least one sample of all commutation rates has been maintained.

B.4.1.4.2.2 Specific Data Contents

The channel number, as before, should be encoded in the channel.

B.4.1.4.3 Titan Data

B.4.1.4.3.1 Discussion

The Titan format is unique to this study in that super-commutation instead of sub-commutation is used. See Figure 3.3-4 (Titan Launch Vehicle Commutation Format) and Table 3.3-5 (not included). In order to simplify the data encoding, the Commutation Format has been modified to a sub-commutated version, see Figure B-3-4-5. A check of this against Figure 3.3-5 will show the results to be identical. A word should be read from each of -3,-4, and -5 in turn and thru recycle. For example, 177 on-3 178 on-4, 179 on-5, then back to 180 on-3.

---

* All Figure numbers which do not contain the Appendix B designation refer to those figures in the original specification.

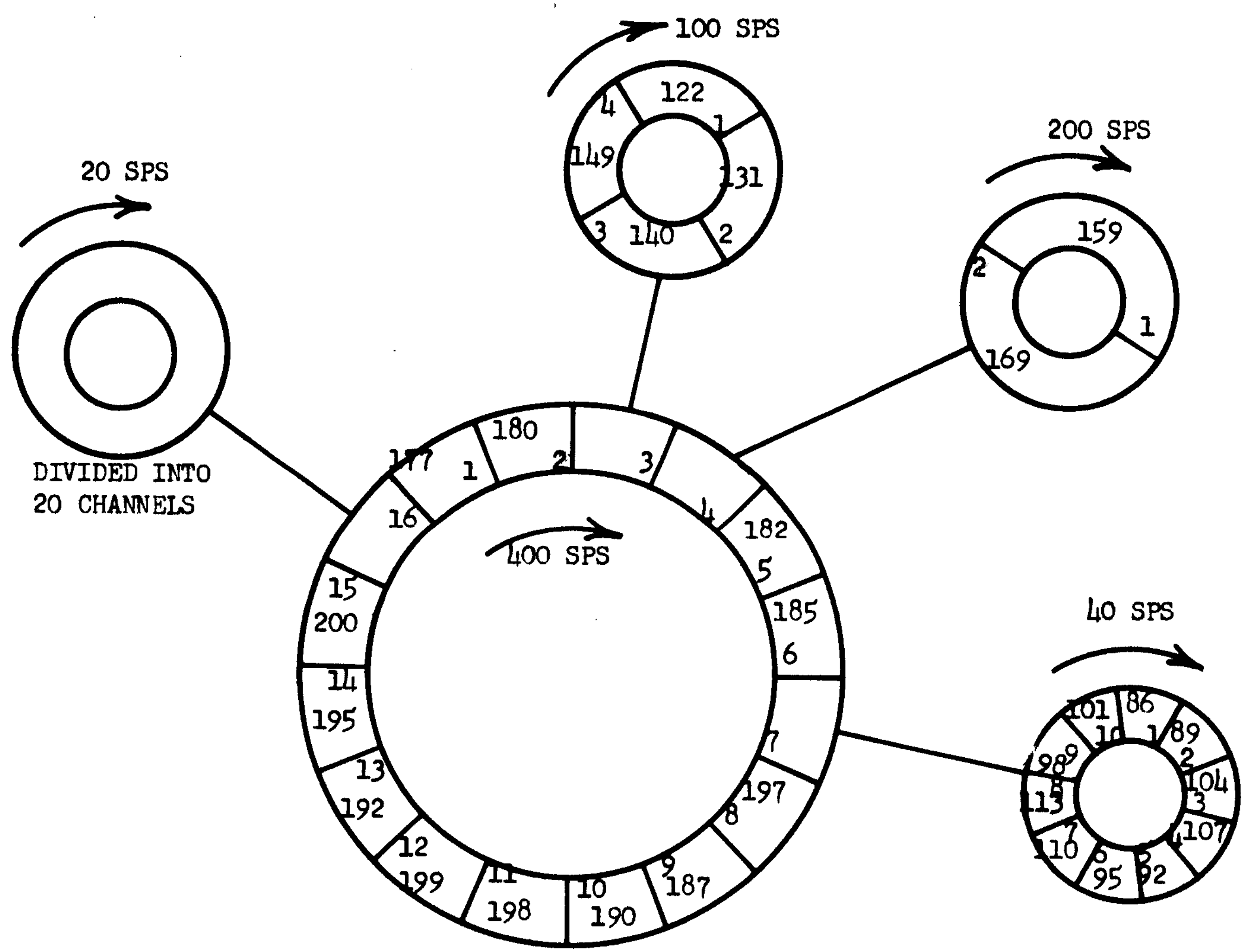


FIG. B-3 FIRST SYLLABLE TITAN COMMUTATION ANALOG
SIMPLIFIED FOR BURST DEMONSTRATION

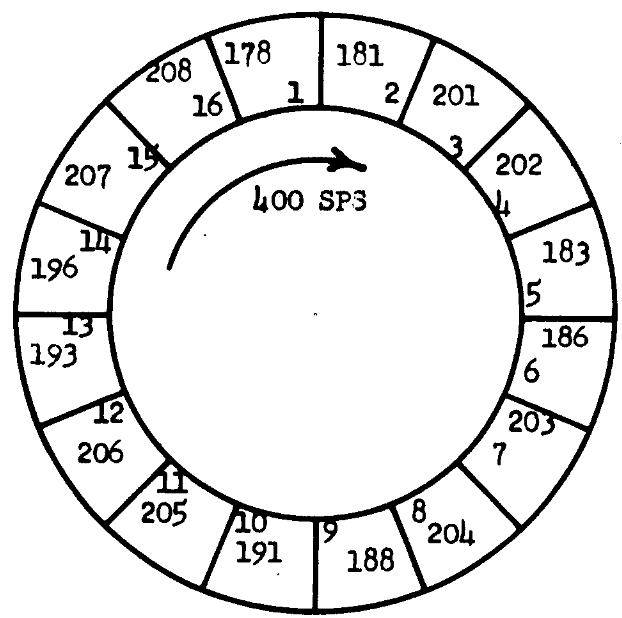


FIG. B-4 SECOND SYLLABLE TITAN COMMUTATION ANALOG SIMPLIFIED FOR BURST DEMONSTRATION

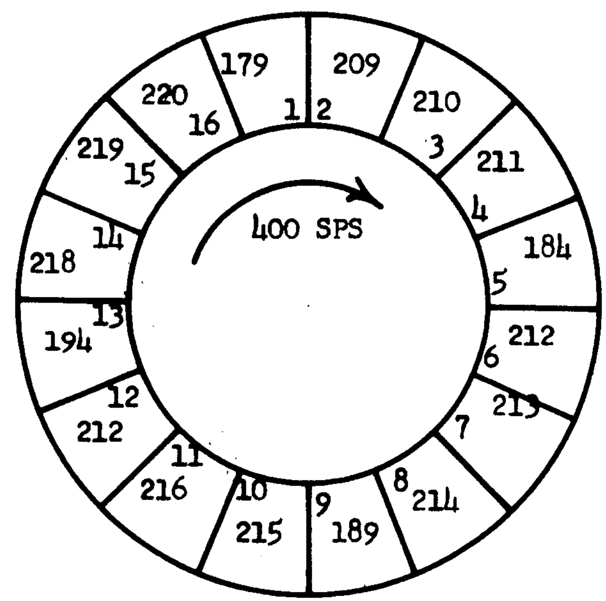


FIG. B-5 THIRD SYLLABLE TITAN COMMUTATION ANALOG SIMPLIFIED FOR BURST DEMONSTRATION

B.4.1.4.3.2 Simplification

Modifying the data pattern to yield one of each sampling rate produces the results of Figure B-3-4-5. New channel assignments, running from 196 through 220, have been inserted in the now vacant channels.

B.4.1.4.3.3 Specific Channel Contents

The channel number should appear in all channels, including the renumbered ones, except channel 190 which will contain a digitized E.C.G. signal.

B-4.2 Parameter Cards

B.4.2.1 Channel Identification Cards

B.4.2.1.1 Description

The cards will give the chosen channel numbers, which could vary from 1-511, and the data source. The actual channel numbers can be picked using the tables in the Appendix on channel addressing. (The correct channel mix and order of channel call-up must be chosen prior to the test). The time difference between sync and the output of interest will also be listed on the card. This time difference will be used to reconstruct the time word when the tape is replayed. The time differences for each channel can be computed using the Commutation formats, Figures B-1 through B-5. For example, on Gemini, channel 120 occurs:

$$\frac{120 \text{ revolutions}}{160} \quad \frac{1}{40 \text{ revolutions/sec}} = 0.018 \text{ seconds}$$

after sync.

B.4.2.1.2 Data Format

Bit position #	1 - 9	First channel address
	10 -11	Vehicle data source
	12 -28	First channel t_{Diff}
	41 -49	Second channel address
	50 -51	Vehicle data source
	52 -68	Second channel t_{Diff}

B.4.2.2 Timing Variation Cards

The card will have the chosen clock rate, 100---104 bits

Bit position # 1-3 Clock rate

B.4.3 Time of Day Recorder Tape

B.4.3.1 Time of Day Correlation

B.4.3.1.1 Purpose

A simulated time of day is dupped on a tape. This time will be compared with sync from each of the three missile data channels, and a time of missile sync will be derived and stored. This time will then be used at a later time to find the period-by-period time scale and variation.

B.4.3.1.2 Format

The T.O.D. will be in hours, minutes, seconds, and micro-seconds. The number of bits to be used is 37*, which is divided as:

Period	Hours	Minutes	Seconds	Microseconds
Max. No. of Units	23	59	59	10^6-1
Number of Bits	5	6	6	20

B.4.3.1.3 Coding

B.4.3.1.3.1 Philosophy

At the maximum rate of data input, 41.7KBPS at 36 bits per data block, approximately 1,158 data blocks will appear per real second (if this data is at the same density as telemetry data), or 9,611 data blocks per data second (at the 8.3:1 scale factor). The precision of the time block is governed by the number of time blocks per data block since the length of a data second has already been fixed by the required sampling rates of the telemetry data. If all 36 bits/data block were used for a time word, the word could be read to a precision of (approximately) 1 second/10K time blocks = 100 microseconds. If only 18 bits were needed, the precision would be 1 second/20K time blocks = 50 microseconds. However, those 18 bits would have to be proportioned to give 15 bits for microseconds, to count up to 1,000K microsecond/50microseconds = 20K and only 3 bits for seconds. The second code would have to recycle every 4 seconds.

Table B-2

4 Time Bits Used in 36 Bit Data Word	Precision of Time	Related Time Inc.	Breakdown (This Demons) Hrs.	Min.	Sec.	uSec.	Legend
36 $\frac{\text{1 Time Word}}{\text{1 Data Word}}$	100	10K	5	6	6	19*	*Not all needed

*For this demonstration only 36 bits will be used. Therefore, the maximum number of hours to be used will be 15.

Table B-2 (Cont'd)

18 = $\frac{\text{2 Time Word}}{\text{1 Data Word}}$	50	20K	-	-	3	15
24 = $\frac{\text{3 Time Word}}{\text{2 Data Word}}$	66 2/3	15K		4	6	14
27 = $\frac{\text{4 Time Word}}{\text{3 Data Word}}$	75	13.3K	1	6	6	14

Timing Bit Breakdown

The precision of the time error which will be detected (in the interlaced system) will be:

$$\pm \frac{1 \text{ bit}}{102 \text{ bit period}} \times 50{,}000 \text{ microseconds} \cong 500 \text{ microseconds}$$

Note that in real system precision 0.05% x 50,000 microseconds - 25 microseconds. Since even the coarsest unit, i.e., one time word per data word, gives a precision five times that of the minimum error precision, that will be used.

B.4.3.1.3.2 Mechanization

In order to keep the time scaling linear between points, a more exact precision which approaches 104 microseconds per block must be used.

For example:

Block Number	Time Period
1	$10^h:3^m:23^s$:123,400
2	:123,504
3	:123,608
4	:124,712

B.5 Output Data

B.5.1 Output Tape:

B.5.1.1 General

The output data tape represents the merged outputs of the three input tapes. As such it must be responsive in some fashion to the input data rates, for if the rate falls below the specified rate (of 40.8KBPS scaled), the memory will "overflow". If the rate exceeds the scaled output data rate, care must be taken, especially in the blocked system, that readout does not occur before memory read-in load in reaches the specified point.

B.5.1.2 Scaling

For the Interlaced System - the slot sizes vary between 29 and 37 bits with some 40 bit entries. Therefore, a standard 36 bit word will be adapted to the slot size. The 37 bits and 40 bit entries will be shortened. Since three slots make up a standard 102 Interlaced Format period,

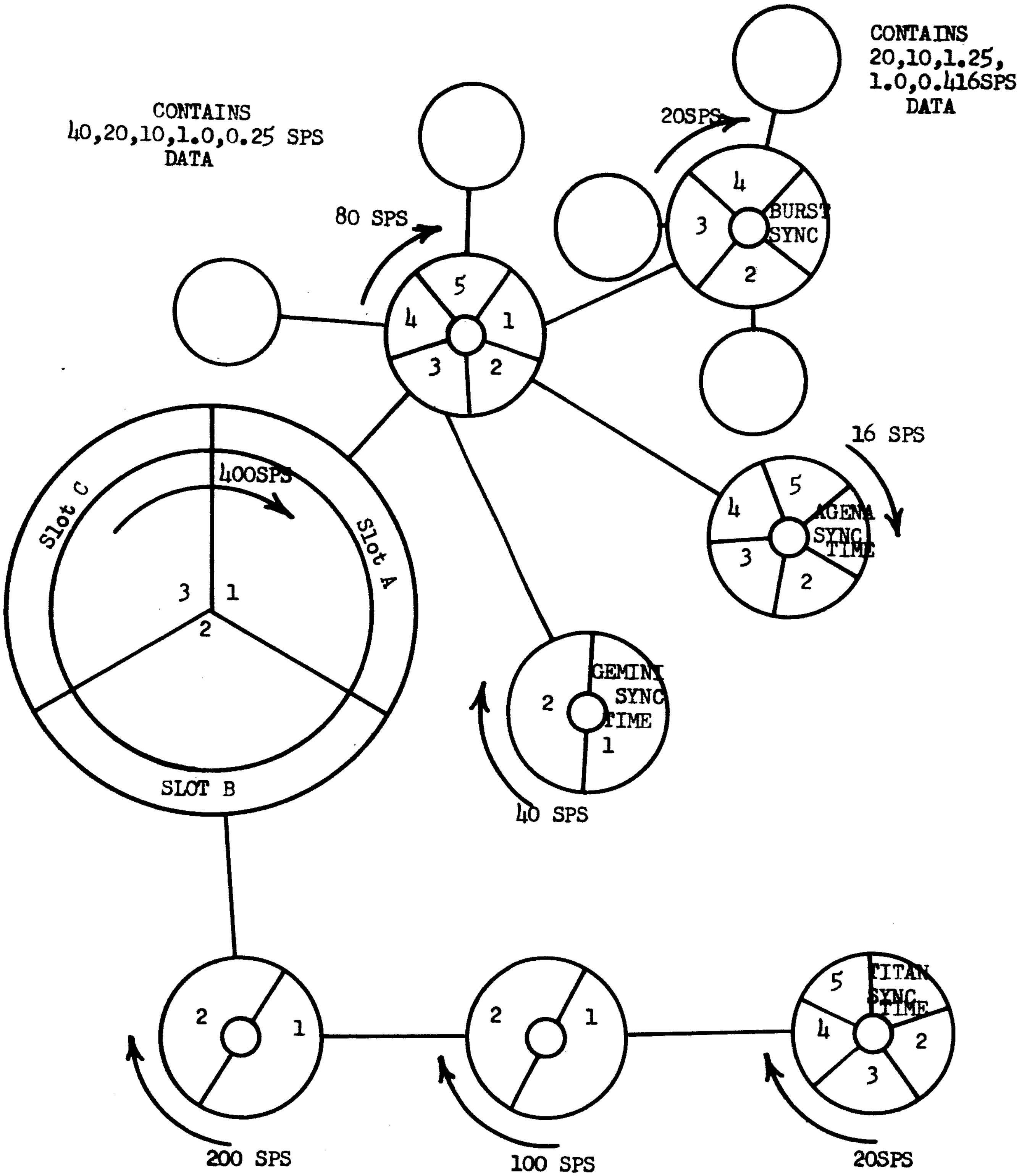


FIG. B-6 PERIODIC BURST COMMUTATION FORMAT three standard (108 bit) data blocks will be equal to this. Since both the input and output tape does have the same data capacity, the output data scaling is:

$$(8.3 \text{ input data word scaling} \quad \frac{153.6 - \text{mod.input Titan rate}}{40.8 - \text{output data rate}} \quad \frac{102 \text{ bits}}{108 \text{ bits}} \quad \frac{1 \text{ data block}}{2 \text{ input data words}}$$

= 14.7 and only one out of every 15 output blocks is data.

B.5.1.3 Coding

In the final-type system, a number of control signals would have to be sent at some time which would identify some significant point in the data message so that points, such as sync, could be identified. In this demonstration, no such command exists and, therefore, coding to identify specific data blocks will be assigned. Since the sub-commutated channels of the Agena and Gemini carry their own sub-frame identification, it will only be necessary to identify the twenty period between Titan sync. Therefore, a code ranging from 1 to 20 (5 bits) will be used to identify the 20 data blocks (0 will be reserved for non-data blocks).

B.6 Program Operation

B.6.1 Time of Day Correlations

The first bit of (Titan) sync will be compared with the specific T.O.D. This information should then be sent to memory. Only the last 18 LSD bits of Agena and Gemini will be used in the Burst system. As each additional time of sync occurs it should be compared with the previous time of sync, and this difference minus the standard period length will be stored as the deviation.

Titan deviation - T_n - t_{n-1} - 1/20 in seconds

Gemini deviation- T_n - T_n-1 - 1/40 in seconds

Agena deviation - T_n - T_n-1 - 1/16 in seconds

B.6.2 Sync Counter

Successive sync pulses are to be counted in order to establish the format and the sub-frame rates:

B.6.2.1 Titan

A 5 bit counter is strobed by the Titan sync. The counter is capable of counting the twenty sync frames which occur in a second.

B.6.2.2 Gemini

A 7 bit counter will count the 96 consecutive occurrence of Gemini sync. The information in this counter could be compared with the prime sub-frame address. For example:

Sync Decimal	Counter Binary MSD LSD	Prime Sub-Frame Decimal	Address Binary
95	1011111	23	10111

B.6.2.3 Agena

A 7 bit counter will be enabled to count the 80 sync pulses which determine the sub-frame ratio.

B.6.3 Internal Sync Generator

An internal oscillator generates a sync pattern which precedes each data period and identifies the start of this period.

Pattern - A 24 bit sync pattern is generated every 1/20 second. The pattern is:

5 4 3 New bits

000001101101011110111001

The sync Burst oscillator is initially triggered by the Titan sync (delayed) and thereafter free-runs.

B.6.4 Address Comparator

Addresses from the parameter cards are compared with those on the tape. The selected data is sent to memory.

B.6.5 Data Extraction and Storage

As each piece of data is derived from each of the three tapes, it should be placed in memory. (It will be necessary to have recycling memories for both systems and we may want to have specific storage spots for each piece of data.)

B.6.6 Frame Development

B.6.6.1 Frame Size, Normal

Normally the data output should be arranged as:

B.6.6.1.1 First Period, 21, 41, etc.

(a) Slot a - 29 bits
(b) Slot b - 40 bits
(c) Slot c - 33 bits, including guard bit
102 bits

B.6.6.1.2 All Other Periods - 2 thru 20, etc.

(a) Slot a - 37 bits
(b) Slot b - 32 bits
(c) Slot c - 33 bits
102 bits

B.6.6.2 Frequency Varying Case

The capability should be included for varying the length of slot c by ± 1 bit (about $\pm$ 1%).

B.6.6.3 Parameter selection

For purposes of the demonstration, the psuedo 80SPS and 20SPS channel data which is to be in slot A will be specified in terms of the acceptable channel mixes of 40, 20, 10, 1.25, 1.0 and 0.416SPS. Data as follows: (See Output Format Figure):

Data			Comment
First Psuedo-80SPS Channel			Divided into 4-20SPS Channels (Not really)
"	20SPS	"	Burst Sync (Not really involved)
Second	20SPS	"	24-0.416SPS Channels - 1 -10SPS Channel
Third	20SPS	"	8 - 1.25SPS Channels - 1 -10SPS Channel
Fourth	20SPS	"	1 - 20SPS Channel
2nd, 3rd Psuedo - 80SPS Channels			Already Specified
4th Psuedo - 80SPS Channel			20 - 1.0SPS Channels
			3 - 20SPS Channels
5th Psuedo - 80SPS Channels			2 - 40SPS Channels

The number of sub-commutated channels which are presently encoded are:

Sampling Rate SPS	Number of Channels
0-416 (5/12)	24
0.2	80
1.0	128
1.25	8
10	2
20(Titan)	20
20	Not Encoded
40(Titan)	10
40	159

If delay between Titan sync is greater than 581 microseconds and less than 2250 microseconds after Burst sync:

(a) Load time of Titan sync into slot 1b.
(b) Load all called for slots into memory.
(c) Switch memory and readout 4 - 400 SPS data into slot 1c.

Load other channels similarly.

B.6.8 Data Sequencing and Retransmission

a. Sense internal Burst sync and insert into slot 1a.
b. Titan data - load parameters as follows:
   (1) Time of Titan sync into 1b.
   (2) Insert other parameters as demonstrated in Data Format.
c.. Agena data - as under b.
d. Gemini data - as under b.

Table B-3

Slot	Data Word Count		Data Partioning	Bit Counts Bits Used	Block Count Blocks Used
1A	Burst Sync	24	Burst Sync	24	8
	Titan Index	5	Titan Index	5	2
		29		29	10
2A,3A	Time of Agena Gemini Sync	26	Time of Agene Gemini Sync	25	9
	Index	7	Index	7	3
	Extra	4		32	12
		37			
All Other A Slots	4 8-Bit Word	37	4 8-Bit Words	32	12
1B	Time of Titan Sync	37	Titan Sync	36	12
All Other B Slots	8-Bit Words	37	4-8 Bit Words	32	12
All C	4-8 Bit Words	32	3-8 Bit Words	24	9
	Guard Bit	1	1-8 Bit Word Plus Guard Bit	9	3
		33		33	12

All Simulation Output Slots are 36 Bits Long - Blocking of Output Data.

B.7 Data Replay

In deciphering the output data tape, care must be taken to insure that each of the data channels can be separated and interpreted. To accomplish this, the format for both the Interlaced and Blocked will be arranged to conform to the IBM 36 bit word and the actual representation of the twelve bit words that it can be broken into. The organization of this will be as follows:

To read out a particular parameter,(a)Identify Burst sync (one out of every sixty blocks) by its particular pattern;(b)The parameter of interest will be some known number of IBM data blocks (data slots) after Burst sync. Similarly, the parameter can be located within the data block by a key;(c)Time of each parameter can be established using the time of that period and then reconstructing time (as shown in C).

To find out the time at which any one channel was strobed out, to the time of sync should be added the time difference between the sync and the channel readout plus the error in time syncing:

$$t_n = t_{sync} + t_{diff} \left(1 + \frac{t_{dev}}{period}\right)$$

The time of sync and the time deviation is available from memory. The period between syncs in a constant for each channel, 1/20 second for Titan, 1/16 for Agena, and 1/40 second for Gemini. The location of a channel with respect to the total number of channels varies from vehicle to vehicle. This difference parameter should be an input.

The general formula for constructing the time code is:

$$t_n = t_L + t_{Diff} \left(1 + \frac{t_{Dev}}{period}\right)$$

where t_L = the time of occurrence of the lowest sync rate which occurs at 1/20 second intervals for Titan, 0.416 seconds for Agena, 0.20 for Agena.

$\frac{t_{Dev}}{period}$ = since its the deviation time over one period.

The t_{Diff} term represents the offset from sync and is:

$$t_{Diff} = T_H \frac{S_H}{S} a + b + c$$

Where T_H = the period of highest sampling rate

S_H = highest sampling rate

S_L = lowest sampling rate

S = sampling rate of channel of interest

a = variable multiplier

$0 \quad a \quad (S/S_L - 1)$

b = sub-frame number minus 1 of channel of interest

$0 \quad b \quad (S/S_H/S - 1)$

c = location in parts of highest rate revolution of channel of interest

Example: Assume that we want to readout channel 169, a 200SPS channel which is in the second sub-frame number in the fourth of 16 channels (in the 400SPS commutation format).

$$t_{Diff} = TH \left(\frac{SH}{S} a + b + c\right)$$

$$= \frac{1}{400} \quad \frac{400SPS}{200SPS} \quad a + (2-1) \text{ sub-frame} + \frac{4 \text{ channel}}{16 \text{ channel}}$$

$$= \frac{1}{400} \quad (2\ a + 1\ 1/4) \qquad 0 \quad a \quad 9$$

$$\text{and } t_{Diff} = \frac{1}{400} \underset{a=0}{\overset{9}{}} \quad (2\ a + 1\ 1/4)$$

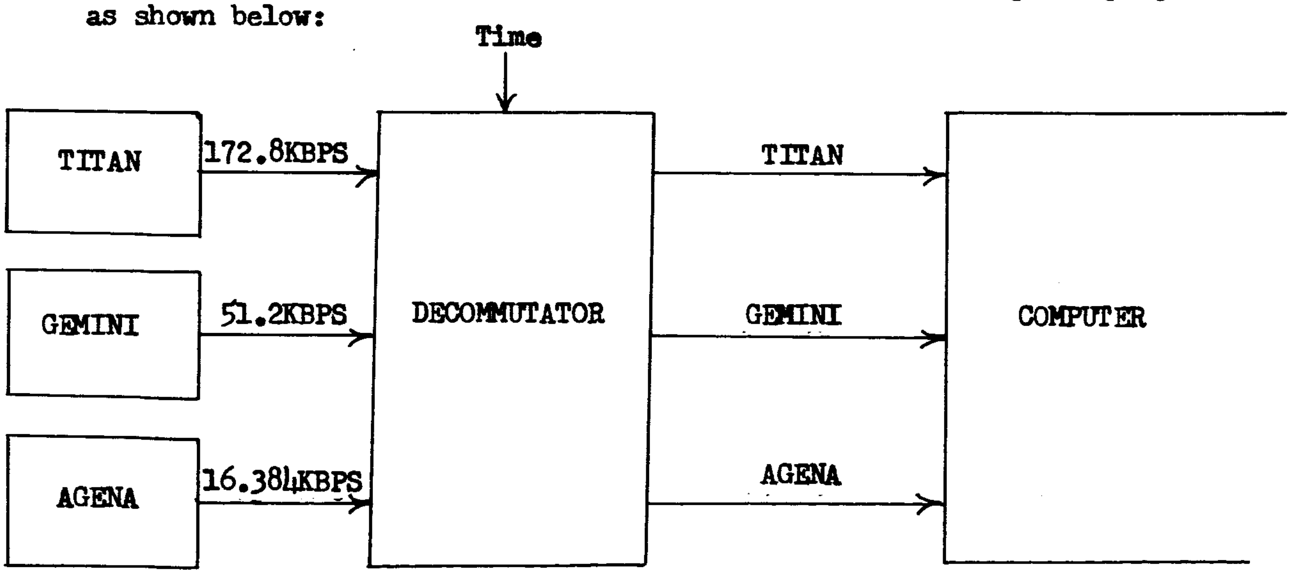
$$t_n = t_L + \frac{(1 + \quad)}{400} \quad \underset{a=0}{\overset{9}{}} \quad (2\ a + 1\ 1/4)$$

where the time (t_L) of the lowest sync occurences, and = deviation, can be plugged in as defined.

B-8 Simulation of Data Rates

B-8.1 Philosophy

The actual data rates from the three vehicles into the computer proper are as shown below:



Decommutation Philosphy

For purposes of the demonstration, the data handled by the computer will be already assumed to be decommutated and properly addressed for handling in the computer. Therefore, the actual tape state inputs will be derived from the addressed data, which data will have a different rate than the telemetry input (because of addressing). Inasmuch as possible, this data will compare with the input rates.

B.8.2 Data Scaling

The varying data rates for each of the missiles, 172.8K bits per second for Titan, 51.2K bits per second for Gemini, and 16.384K bits per second for Agena can be scaled so that the data rate for Titan is three times that of the Gemini, and that for Agena is one-third of Gemini.

Vehicle	Real Data Rate - KBPS	Scaled Data Rates
Titan	172.8	172.8*
Gemini	51.2	51.2 (ref)
Agena	16.384	17.07

B.8.3 Data Addressing

B.8.3.1 Preferred Technique

The previous Appendix on data addressing has shown a necessity for 9 bits of channel identification with each 8 bits of data word. In addition, two more identification bits must be added at some point to identify which of the three missile sensors the information was derived from. Therefore, a total of 19 bits must be used. In order to parcel the 19 bit code in the best fashion in the standard 36 bit word would require a loss of almost half the available data space. Therefore, a 17 bit code will be adopted which will allow two data words to be compressed in each 36 bit standard word. It will, of course, be necessary to differentiate which sensor the data is read from by some other device.

The format will then be:

Bit Position #	1 - 9	First Word Address
	10 -18	First Data Word
	19 -27	Second Word Address
	28 -36	Second Data Word

Each of the 64 words in the 5 minor frames is composed of 3 eight-bit syllables plus a sync code of 3 bits. A callout of any one word in the Titan format thus requires 10 bits.

* Compensated to correct for three sync bits associated with every 24 bit data word.

B.8.3.2 Second Technique

A second technique could be used if the emphasis were strictly on the faster read-in of data at the expense of programming. This reasoning comes about because the Titan input data link is much the heaviest loaded link and dominates the development. At the same time Titan is unique in that its data word from the missiles is organized as follows: specification of which of the 64 words (6 bits) in any minor frame syllable (4 bits) is specified. The minor frame syllable is made up of the five minor frames multiplied by the three syllables in each word. The 10 address bits combined with the 3 eight-bit syllables give a total of 34 bits out of the 36 bit format and 3/2 as much read-in rate as the Preferred Technique.

The disadvantages of this technique are:

(a) The address specification for Titan becomes unique with respect to the other two missile addresses.
(b) Specification of any one word from Titan becomes a more complex operation of specifying a minor frame, a syllable, and a word location instead of just an address as in the Preferred Technique.

B.8.4 Data Rate Scaling

Using the preferred technique, $\frac{(41{,}700 \text{ bits/sec})}{36 \text{ bits/block}} \; \frac{2 \text{ data words}}{1 \text{ standard block}} =$

$2316\ ^{2}/_{3}\ \frac{\text{data words}}{\text{second}}$ which is equivalent to $(2316^{2}/3\ \frac{\text{data words}}{\text{second}}$

$(\ \frac{27 \text{ bits}}{3 \text{ data words}}\ ) = 20.85\text{K}$ bits/second or a scale-down ratio of 8.3 to 1

from the actual 172.8K bits/second Titan rate. (The restriction on scaling of data rates is on the tape drive. The actual system would have a specific input data buffer, and the restriction would then be on the computer clock rate which is sufficient to handle the actual 172.8K bits/second input rate.)

Note that the second technique would lower the 8.3 scale-down rate by a third.

B.8.5 Length of Data Runs

At 75 inches/second a tape 2500 ft long will take 384 seconds to run. The data scaling factor of 8.3:1 scales this to about 45 seconds. Therefore, the highest sampling rate (400SPS) will contain 18K samples, the lowest (0.25SPS) about 180 samples.

B-9 Data Channel Addressing

For output frame-by-frame data from the program, it will be necessary to satisfactorily identify the sub-commutation ratios of 80:1 (Agena) and 96:1 (Gemini) in the output data formats. Similarly, at least partial sub-frame identification is on the telemetry link itself (although no sub-frame identification will appear on the simulated telemetry tape which has been assumed to be decommutated and addressed). For purposes of data handling within the computer program proper, however, handling will be on a word-by-word process, and each word must carry its own identification. Therefore, to minimize the number of identification bits which each word must carry, the sub-frame addressing will be dropped (internally) and a complementary identification will be used. Illustrations will follow. A saving of 5 bits in the address results from this technique.

A table correlating the channel assignments with the proper identification follows:

Agena

Referring to Figure 3.2.12, the sub-commutation format (mechanical analog) of the Agena Multiplexer, it is apparent that a matrix of the 128 (7 bit) main frame channels squared with the 80:1 (7 bit) sub-commutation ratio would require a 14 bit identification. However, the total number of channels are:

Main Frame	128 channels
Sub Frame A	128
Sub Frame B	240*
Sub Frame C	16
	512 channels = 9 bits

Note that for main frame channel correlation there is a one-to-one relationship, and for sub-frame A, a similar relationship exists after taking the 128 factor into consideration.

Titan

Referring to Figure 3.3.5, Major Frame Format for Tital Telemetry, a matrix of 64 words (6 bits) by 15 minor frame words (4 bits) would require 10 ID bits. Because the Titan format does not use a sub-commutation ratio, however, many of the assignments are redundant. The actual number of channels used are:

196	analog channels
6	combined bi-level channels
5	sync words
207	words = 8 bits

The Titan word is 27 bits long and composed of three 8-bit syllables plus three sync bits at the end. All sync bits are the same, "110", except the main sync "001".

---

* Each word is equivalent to three 8-bit words.

Table B-4

Frame	Channel Number (Decimal)	Supplemental Channel Number (Decimal)
Main		
	128	128
Sub-Frame A		129
	128	256
Sub-Frame B	17 - 1	257
	17 -80	336
	18 - 1	337
	18 -80	416
	19 - 1	417
	19 -80	496
Sub-Frame C		497
	16	512

Channel Correlation Table - Agena

The channel correlation follows:

Table B-5

	Channel Number (Decimal)	Supplementary Channel Number (Decimal)
Analog	1	1
	196	196
	B-1	197
	B-5	201
	B-6	202
	Sync 1	203
	Sync 4	206
	Main Sync	255

Channel Correlation Table - Titan

Gemini

It will be obvious, after reference to the Gemini Commutation Format, Figure 3.2.4, that the Gemini case is similar to the Agena case. The matrix formed by 160 main frame (8 bit) words by a sub-commutation ratio of 96:1 (7 bits) would force a high of 15 address bits. The total number of channels is:

Main Frame (40SPS)	160	Channels
First Sub-Frame (10SPS)	21	"
Second Sub-Frame		
Six - 24 channel (1.25)	144	"
Fourteen - 8 "(0.416)	112	"
	437	channels = 9 bits

In summation, a 9 bit ID code will identify each channel properly.

Table B-6

Main Frame Channel Number (Decimal)	Sub-Frame Channel Number (Decimal)	Second Sub-Frame Number (Decimal)	Supplementary Channel Number (Decimal)
1 Sync			1 Sync (Only one needed)
2 "			2
3 "			3
160			160
14	2	1	161
14	2	8	168
14	3		169
14	4		170
14	4	24	193

Similar assignments could hold for other Channels

Channel Correlation Table

Gemini